

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação

# **Estudo de Desempenho do Sistema 3G 1xEV-DO Através de Modelos Reais de Tráfego**

**Autor: Leandro Bento Sena Marques**

**Orientador: Prof. Dr. Shusaburo Motoyama**

**Dissertação de Mestrado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática** .

Banca Examinadora

Shusaburo Motoyama, Dr. .... DT/FEEC/Unicamp  
Magda Patrícia Caldeira Arantes, Dr<sup>a</sup>. .... UNIFIAN  
Michel Daoud Yacoub, Dr. .... DECOM/FEEC/Unicamp

Campinas, SP

Julho/2005

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M348e Marques, Leandro Bento Sena  
Estudo de desempenho do sistema 3G 1xEV-DO através  
de modelos reais de tráfego / Leandro Bento Sena Marques.  
–Campinas, SP: [s.n.], 2005.

Orientador: Shusaburo Motoyama  
Dissertação (Mestrado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Elétrica e de  
Computação.

1. Telecomunicações - Tráfego. 2. Sistemas de  
comunicação sem fio. 3. HTTP - (Protocolo de redes de  
Computadores). 4. Protocolo para transferência de arquivos  
(Protocolo de redes de Computadores). 5. Protocolo para  
aplicação de rede sem fio (Protocolo de redes de  
Computadores). I. Motoyama, Shusaburo. II. Universidade  
Estadual de Campinas. Faculdade de Engenharia Elétrica e  
de Computação. III. Título

Título em Inglês: Performance study of the 3G 1x-EV-DO system through traffic real models.

Palavras chaves em Inglês: Real Traffic, HTTP, FTP, WAP, schedulers FIFO, with priority, WFQ  
e DRR.

Área de Concentração: Telecomunicações e Telemática.

Titulação: Mestre em Engenharia Elétrica

Banca Examinadora: Magda Patrícia Caldeira Arantes e Michel Daoud Yacoub.

Data da Defesa: 15/07/2005.

# Resumo

Neste trabalho é apresentado um estudo do desempenho do enlace direto do sistema 1xEV-DO. O estudo enfatiza o impacto da adoção de modelos de tráfegos reais tais como HTTP, FTP e WAP e de diferentes escalonadores. Através de simulação são analisados: o atraso médio dos pacotes, a vazão e percentual de perda de cada tipo de tráfego em função da carga de tráfego. É adotado neste estudo o escalonamento FIFO, com prioridade, WFQ e DRR. Os resultados mostram que o sistema 1xEV-DO padronizado permite transmissões de dados a altas taxas de bits. É mostrado, também, que a introdução de outros escalonadores podem melhorar ainda mais o desempenho do sistema.

**Palavras-chave:** Modelos reais de tráfego HTTP, FTP, WAP, escalonamentos FIFO, com prioridade, WFQ e DRR.

# Abstract

The forward link performance study of the 1xEV-DO system is presented in this work. The study emphasizes the impact of the adoption of the real traffic models such as HTTP, FTP and WAP and several different schedulers. The average packet delay, the throughput and each traffic loss percentage are analyzed through simulation in function of traffic load. It is adopted in this study schedulers FIFO, with priority, WFQ and DRR. The results show that standardized 1xEV-DO system allows data transmission at high bits rates. It is also shown that the system performance can still enhanced by introducing others schedulers.

**Keywords:** Real traffic, HTTP, FTP, WAP, schedulers FIFO, with priority, WFQ and DRR.

*Aos meus pais, irmã, avós e tios*

*"Bem-aventurado o homem que encontra sabedoria, e o homem que adquire conhecimento, pois é mais valiosa do que ouro e prata." (Provérbios 3,13)*

# Agradecimentos

Ao meu orientador, Prof. Shusaburo Motoyama, sou grato pela orientação.

Aos colegas Edilson Fernandes De Arruda, Fabio Hernandez, Vinicius Foletto Montagner, Ricardo Coelho Silva, Rúbia Mara De Oliveira, Tatiane Regina Bonfim e Alessandro Do Nascimento Vargas pelas sugestões de modelos de tese.

Aos demais colegas do departamento de telemática, pelas críticas e sugestões.

À minha família e à minha namorada Débora pelo apoio durante esta jornada.

A Deus pela saúde e pela oportunidade de concluir este mestrado.

À CAPES, pelo apoio financeiro.

# Sumário

<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xv</b>
<b>Glossário</b>	<b>xvii</b>
<b>Lista de Símbolos</b>	<b>xxi</b>
<b>Trabalhos Publicados Pelo Autor</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 O Sistema IS-856 (1xEV-DO)</b>	<b>7</b>
2.1 Introdução . . . . .	7
2.2 Arquitetura da Rede . . . . .	7
2.3 Protocolos de Rede . . . . .	9
2.4 Interface Aérea - Enlace Direto . . . . .	10
2.4.1 Formato do Quadro . . . . .	11
2.4.2 Taxas de Transmissão . . . . .	12
2.5 Escalonador de Equidade Proporcional . . . . .	13
2.6 Soft Handoff Virtual . . . . .	14
2.7 Redundância Incremental . . . . .	14
2.8 Interface Aérea - Enlace Reverso . . . . .	15
2.9 Conclusão . . . . .	16
<b>3 Modelos Reais de Tráfego, Escalonadores e Modelagem do Sistema IS-856</b>	<b>17</b>
3.1 Introdução . . . . .	17
3.2 Modelo TCP . . . . .	17
3.3 Modelo HTTP . . . . .	21
3.3.1 Características do Modelo de Tráfego HTTP . . . . .	21
3.3.2 Parâmetros do Modelo de Tráfego HTTP . . . . .	22
3.3.3 Histogramas das Fontes HTTP . . . . .	24
3.4 Modelo FTP . . . . .	24
3.4.1 Histogramas das Fontes FTP . . . . .	26
3.5 Modelo WAP . . . . .	28

3.5.1	Histogramas das Fontes WAP . . . . .	29
3.6	Escalonadores . . . . .	30
3.6.1	FIFO . . . . .	31
3.6.2	Prioridade . . . . .	32
3.6.3	WFQ . . . . .	32
3.6.4	DRR . . . . .	36
3.7	Modelagem do Sistema IS-856 . . . . .	37
3.7.1	Modelo IS-856 . . . . .	37
3.7.2	Cenários de Análise . . . . .	39
3.8	Conclusão . . . . .	41
<b>4</b>	<b>Análise dos Resultados</b>	<b>43</b>
4.1	Introdução . . . . .	43
4.2	Cenário 1 . . . . .	43
4.3	Cenário 2 . . . . .	45
4.4	Cenário 3 . . . . .	46
4.5	Cenário 4 . . . . .	47
4.6	Cenário 5 . . . . .	49
4.7	Modelo Teórico Para o Tráfego Poissoniano Sem Prioridade . . . . .	54
4.8	Conclusão . . . . .	58
<b>5</b>	<b>Conclusões</b>	<b>59</b>
	<b>Referências bibliográficas</b>	<b>61</b>
<b>A</b>		<b>65</b>
A.1	Consistência do Modelo IS-856 Desenvolvido . . . . .	65
A.2	Histogramas das Fontes Poissonianas . . . . .	68
<b>B</b>		<b>73</b>
B.1	Programa Principal . . . . .	73
B.2	Fonte HTTP . . . . .	74
B.2.1	HTTP Burst 1.0 . . . . .	75
B.2.2	HTTP Persistente 1.1 . . . . .	78
B.3	Fonte FTP . . . . .	80
B.4	Fonte WAP . . . . .	81

# Lista de Figuras

1.1	Evolução dos padrões de telefonia celular . . . . .	4
2.1	Arquitetura de Rede IS-856 . . . . .	8
2.2	Protocolos de Rede IS-856 . . . . .	9
2.3	O sistema IS-856 Maximiza a Vazão via Eficiente uso da Potência [1] . . . . .	11
2.4	Protocolos de Rede IS-856 . . . . .	12
2.5	Pacote de 2 TSs com Intervalo de Transmissão de 4 TSs . . . . .	15
3.1	Segmentos de Controle em uma Conexão TCP . . . . .	18
3.2	Controle de Fluxo TCP . . . . .	19
3.3	Processo de Chegada de Pacotes na ERB . . . . .	20
3.4	Seção de Navegação Web . . . . .	21
3.5	Uma Típica Página Web e seu Conteúdo . . . . .	22
3.6	Modelagem de <i>Download</i> de uma Página Web . . . . .	24
3.7	Histograma do Comprimento do Pacote HTTP . . . . .	25
3.8	Ordem de Chegadas dos Pacotes do Tráfego HTTP . . . . .	25
3.9	Histograma dos Intervalos entre Chegadas do Tráfego HTTP . . . . .	26
3.10	Típica Seção FTP . . . . .	26
3.11	Modelo de Tráfego FTP . . . . .	27
3.12	Histograma do Comprimento do Pacote FTP . . . . .	27
3.13	Ordem de Chegadas dos Pacotes do Tráfego FTP . . . . .	28
3.14	Histograma dos Intervalos entre Chegadas do Tráfego FTP . . . . .	28
3.15	Modelo de Tráfego WAP . . . . .	29
3.16	Histograma do Comprimento do Pacote WAP . . . . .	30
3.17	Ordem de Chegada dos Pacotes do Tráfego WAP . . . . .	31
3.18	Histograma dos Intervalos entre Chegadas do Tráfego WAP . . . . .	31
3.19	Pseudo Código WFQ . . . . .	33
3.20	Ciclo 1 WFQ . . . . .	34
3.21	Ciclo 2 WFQ . . . . .	34
3.22	Ciclo 3 WFQ . . . . .	34
3.23	Ciclo 4 WFQ . . . . .	35
3.24	Ciclo 5 WFQ . . . . .	35
3.25	Ciclo 6 WFQ . . . . .	36
3.26	Ciclo 7 WFQ . . . . .	36

3.27	Ciclo 8 WFQ . . . . .	36
3.28	Modelo IS-856 . . . . .	37
4.1	Vazão em Função da Ocupação do Sistema para o Cenário 1 . . . . .	44
4.2	Atraso Médio em Função da Ocupação do Sistema para o Cenário 1 . . . . .	45
4.3	Vazão em Função da Ocupação do Sistema para o Cenário 2 . . . . .	46
4.4	Atraso Médio em Função da Ocupação do Sistema para o Cenário 2 . . . . .	46
4.5	Vazão em Função da Ocupação do Sistema para o Cenário 3 . . . . .	47
4.6	Atraso Médio em Função da Ocupação do Sistema para o Cenário 3 . . . . .	48
4.7	Vazão em Função da Ocupação do Sistema para o Cenário 4 . . . . .	48
4.8	Atraso Médio em Função da Ocupação do Sistema para o Cenário 4 . . . . .	49
4.9	Vazão HTTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	50
4.10	Vazão WAP em função da Ocupação do Sistema para o Cenário 5 . . . . .	50
4.11	Vazão FTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	51
4.12	Atraso Médio das fontes HTTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	52
4.13	Percentual de perda de pacotes HTTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	52
4.14	Atraso Médio das fontes WAP em função da Ocupação do Sistema para o Cenário 5 . . . . .	53
4.15	Percentual de perda de pacotes WAP em função da Ocupação do Sistema para o Cenário 5 . . . . .	53
4.16	Atraso Médio das fontes FTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	54
4.17	Percentual de perda de pacotes FTP em função da Ocupação do Sistema para o Cenário 5 . . . . .	54
4.18	Comparação entre o Modelo Teórico e a Simulação . . . . .	56
4.19	Atraso Médio em função da Ocupação do Sistema com o modelo analítico para o Cenário 4 . . . . .	57
A.1	Vazão em Função do Número de Usuários . . . . .	67
A.2	Atraso Médio dos Pacotes em Função do Número de Usuários . . . . .	68
A.3	Fator de Utilização do Sistema Versus o Número de Usuários . . . . .	68
A.4	Vazão em Função do Número de Usuários . . . . .	69
A.5	Atraso Médio dos Pacotes em Função do Número de Usuários . . . . .	69
A.6	Fator de Utilização do Sistema Versus o Número de Usuários . . . . .	70
A.7	Histograma do Comprimento do Pacote do Tráfego Poissoniano . . . . .	70
A.8	Ordem de Chegada dos Pacotes do Tráfego Poissoniano . . . . .	71
A.9	Histograma dos Intervalos entre Chegadas do Tráfego Poissoniano . . . . .	71
B.1	Fluxograma do Programa Principal do Simulador . . . . .	74

# Lista de Tabelas

2.1	Tipos de Modulação por Taxas de Dados - Enlace Direto . . . . .	13
2.2	Tipos de Modulação por Taxas de Dados - Enlace Reverso . . . . .	15
3.1	Parâmetros do Modelo de Tráfego HTTP . . . . .	23
3.2	Funções Densidades das Distribuições . . . . .	23
3.3	Parâmetros do Modelo de Tráfego FTP . . . . .	26
3.4	Parâmetros do Modelo de Tráfego WAP . . . . .	30
3.5	Distribuição de Probabilidade pela Taxa de Dados . . . . .	38
3.6	Cenário 1 . . . . .	40
3.7	Cenário 2 . . . . .	40
3.8	Cenário 3 . . . . .	40
3.9	Cenário 4 . . . . .	41
3.10	Cenário 5 . . . . .	41
A.1	Distribuição <i>Default</i> das Prioridades (Pr) . . . . .	65

# Glossário

16QAM - *16-positions Quadrature Amplitude Modulation*

1G - Primeira Geração de Telefonia Celular

2,5G - Segunda Geração Intermediária de Telefonia Celular

2G - Segunda Geração de Telefonia Celular

3G - Terceira Geração de Telefonia Celular

3GPP - *3rd Generation Partnership Project*

3GPP2 - *Third Generation Partnership Project 2*

8PSK - *8 Phase Shift Keying*

ACK - *Acknolegment*

AMPS - *Advanced Mobile Phone Service*

BF - *Buffer Finito*

BI - *Buffer Infinito*

BPSK - *BiPhase Shift Keying*

BSC - *Base Station Controller*

BTS - *Basestation Transceiver Subsystem*

CDM - *Code Division Multiplexing*

CDMA - *Code Division Multiple Acess*

CDMA 1xEV-DO - *CDMA 1x Evolution Data Only*

CDMA 1xEV-DV - *CDMA 1x Evolution Data and Voice*

CRC - *Cyclic Redundancy Check*

D-AMPS - *Digital Advanced Mobile Phone Service*

DRC - *Data Rate Control*

DRR - *Deficit Round Robin*

EDGE - *Enhanced Data Rates for GSM Evolution*

ERB - *Estação Rádio Base*

EUA - *Estados Unidos da América*

FIFO - *First In First Out*

FM - *Modulação em Frequência*

FTP - *File Transfer Protocol*

GPRS - *General Packet Radio Service*

GPS - *Global Positioning System*

GSM - *Global System for Mobile Communications*

HTML - *Hyper Text Markup Language*

HTTP - *Hypertext Transfer Protocol*

IMT-2000 - *International Mobile Telecommunications 2000*

IP - *Internet Protocol*

ITU - *International Telecommunications Union*

MSC - *Mobile Switching Center*

MTU - *Maximum Transmission Unit*

NAS - *Network Access Server*

PCF - *Packet Control Function*

PDSN - *Packet Data Serving Node*

PPP - *Point-to-Point Protocol*

Ps - *Poissoniano*

QoS - *Qualidade de Serviço*

QPSK - *Quadrature Phase Shift Keying*

RF - *Rádio Frequência*

RLP - *Radio Link Protocol*

SINR - *Signal-to-Interference-and-Noise-Ratio*

SPr - Sem Prioridade

TCP - *Transmission Control Protocol*

TCP/IP - *Transmission Control Protocol/Internet Protocol*

TDM - *Time Division Multiplexing*

TDMA - *Time Division Multiple Access*

TL - Tempo de Leitura

TS - *Time-Slot*

UDP - *User Datagram Protocol*

UMTS - *Universal Mobile Telecommunications System*

WAP - *Wireless Application Protocol*

WCDMA - *Wideband Code Division Multiple Access*

WFQ - *Weighted Fair Queuing*

# Lista de Símbolos

- $\tau_c$  - representa a soma dos seguintes tempos: o tempo levado por um pacote ACK propagar do cliente para o servidor, mais o tempo de processamento deste pacote pelo servidor e o tempo gasto pelo segmento de dado TCP propagar do servidor para a ERB.
- $\tau_1$  - o tempo de transmissão de um segmento de dado TCP da ERB para o dispositivo móvel por meio da interface aérea.
- $\tau_{total}$  - soma dos tempos  $\tau_c$  e  $\tau_1$ .
- $\lambda$  - taxa de chegada dos pacotes (pacotes/s).
- $\lambda_k$  - taxa de chegada de cada classe k (pacotes/s).
- $\mu$  - taxa de atendimento dos pacotes (pacotes/s).
- $\mu_k$  - taxa de atendimento de cada classe k (pacotes/s).
- $E\{T\}$  - tempo médio total gasto no sistema pelo pacote.
- $E\{W\}$  - tempo médio de espera na fila.
- $E\{W_p\}$  - tempo médio de espera na fila dos pacotes de prioridade p.
- $E\{X\}$  - tempo médio de serviço.
- $\rho$  - ocupação total do sistema.
- $\rho_k$  - ocupação ou carga do usuário k.
- $E\{T_0\}$  - tempo residual de atendimento quando da chegada de um pacote.
- $E\{R_k^2\}$  - valor quadrático médio do tempo de serviço.

# Trabalhos Publicados Pelo Autor

1. L.B.S. Marques, S. Motoyama. “Estudo de Desempenho do Enlace Direto do Sistema 3G 1xEV-DO Utilizando Modelos de Tráfegos Reais”. *XXV Congresso da Sociedade Brasileira de Computação - IV Workshop em Desempenho de Sistemas Computacionais e de Comunicação (SBC - Wperformance'05)*, São Leopoldo, Rio Grande do Sul, Brasil, pg. 2868 - 2885, Julho 2005.

# Capítulo 1

## Introdução

A origem do sistema de telefonia móvel foi impulsionada pela crescente demanda de acesso à informação em qualquer lugar e a qualquer momento. Esta crescente demanda surgiu em razão da telefonia fixa estar atrelada ao acesso fixo e com cabo impossibilitando a mobilidade de seus usuários. Dessa forma, a telefonia móvel inicialmente viabilizava apenas a transmissão de voz com mobilidade, porém com o passar do tempo outros tipos de tráfegos como dados e multimídia associados à mobilidade tornaram-se tão essenciais quanto o tráfego de voz.

A primeira geração de telefonia celular (1G) utiliza modulação em frequência (FM) analógica e a arquitetura de múltiplo acesso FDMA (*Frequency Division Multiple Access*) para a transmissão dos sinais de voz. Mas em razão de esta geração utilizar de forma ineficiente o espectro disponível, logo a capacidade desta foi esgotada.

Nos Estados Unidos da América (EUA), no Brasil e alguns outros países esta geração foi representada na década de 80 pelo sistema móvel AMPS (*Advanced Mobile Phone Service*) proposto pela AT&T e mantido em todo país por uma única operadora. Em contraste com a Europa em que cada país adotou um sistema móvel próprio dificultando o *roaming* internacional [2].

Com o avanço da tecnologia digital surgiu a segunda geração de telefonia celular (2G). Esta geração provê uma melhor eficiência espectral, emprega a tecnologia digital nas transmissões de voz e transmite dados a baixa velocidade [3]. No entanto, esta geração ainda faz uso da tecnologia de comutação de circuitos. A segunda geração possui como representantes as tecnologias CDMA (*Code Division Multiple Access*), TDMA (*Time Division Multiple Access*) e GSM (*Global System for Mobile Communications*).

A tecnologia CDMA (*Code Division Multiple Access*) foi introduzida em sistemas celulares por volta de 1990 com o padrão IS-95. Esta técnica permite o uso de uma banda de 1,25 MHz entre vários usuários simultaneamente por meio de códigos ortogonais entre si [4]. O sistema IS-95 evoluiu significativamente na última década resultando nas revisões IS-95A e IS-95B [5] também conhecidas

como cdmaOne. O padrão IS-95A têm como principais características a comutação de circuitos para voz e o suporte a taxa de dados de até 14,4 kbps. Enquanto o padrão IS-95B atinge uma taxa de dados de até 115 kbps.

Nos sistemas celulares digitais com arquitetura TDMA (*Time Division Multiple Access*), o D-AMPS (*Digital AMPS*) utiliza o padrão IS-136, em que três canais compartilham uma portadora de largura de 30 kHz, com taxa de transmissão total de 48,6 kbps o que implica em 16,2 kbps por canal [6]. Este compartilhamento é dado por intervalos de tempos chamados de TS (*Time-Slot*). A estrutura de transmissão de dados é implementada através de um quadro de 40 ms com 6 TSs de tempo com 6,66 ms cada TS.

O padrão GSM (*Global System for Mobile Communications*) surgiu comercialmente na Europa, no início dos anos 90, como uma alternativa de unificação das tecnologias de telefonia celular. Atualmente este é o padrão 2G mais difundido no mundo. Este padrão caracteriza-se pela utilização da arquitetura de múltiplo acesso TDMA assim como o IS-136. Dessa forma, a banda de 200 kHz é partilhada em 8 TS atingindo taxas de dados de até 9,6 kbps [7]. Além disso, o padrão GSM faz uso da tecnologia de comutação de circuitos.

Embora os sistemas 2G fossem digitais, a capacidade de transmissão de dados era bastante modesta. Com o advento da internet aumentou-se a demanda por transmissões de dados mais altas e levou-se ao aprimoramento da segunda geração de telefonia móvel chamada (2,5G) [8]. Esta geração destacou-se pelo uso da tecnologia de comutação por pacotes como base para a transmissão de dados com taxas mais altas e melhor qualidade de serviço (QoS) em relação a geração 2G. Entre as tecnologias 2,5G de maior destaque têm-se os padrões IS-2000, GPRS e EDGE.

O padrão IS-2000 revisão 0, também conhecido como cdma2000, foi aprovado pelo ITU (*International Telecommunications Union*) em 1999 [5]. Este padrão introduziu a tecnologia de comutação por pacotes na família CDMA permitindo taxas de dados no enlace direto de até 307,2 kbps em uma largura de banda de 1,25 MHz. Além disso, mantém a compatibilidade com as primeiras versões da família CDMA. Entretanto, não era otimizado para o tráfego IP (*Internet Protocol*) de alta velocidade.

O GPRS (*General Packet Radio Service*) é um serviço para comunicação de dados que emprega a comutação de pacotes mantendo a conexão sempre ativa. Este serviço utiliza até os 8 TSs em um canal GSM de largura de banda de 200 kHz e provê uma taxa de dados que teoricamente poderia chegar a 171,2 kbps.

O sistema EDGE (*Enhanced Data Rates for GSM Evolution*) é um padrão desenvolvido para aumentar a taxa de dados para serviços oferecidos pela rede GSM com a mesma banda de 200 kHz. Este aumento é obtido pelas novas características de modulação, codificação e decodificação do sinal, associadas com adaptações do sinal e envio de redundância de informação que aumentam a eficiência da utilização do espectro. É possível desta forma oferecer taxas de dados de até 473,6 kbps [9].

Seguindo a linha de evolução da telefonia celular, chega-se à terceira geração de telefonia celular (3G). Os requisitos para a classificação de um sistema celular como 3G foram definidos pelo IMT-2000 (*International Mobile Telecommunications*). Entre os requisitos destacam-se: as altas taxas de dados de 2 Mbps para ambientes com condições favoráveis, transmissão de dados simétrica e assimétrica, serviços baseados em comutação de circuitos e comutação de pacotes, a melhor eficiência espectral, fornecimento de serviços multimídia e *roaming* global. Assim, os sistemas 3G que atualmente suprem estes requisitos são: o WCDMA (*Wideband Code Division Multiple Access*), o CDMA 1xEV-DO (*1x Evolution Data Only*) e o CDMA 1xEV-DV (*1xEvolution Data and Voice*).

O padrão IS-856 também conhecido como CDMA 1xEV-DO foi otimizado para serviços de dados sem fio de alta velocidade [10]. Este padrão provê taxas de dados de até 2,4576 Mbps no enlace direto (estação rádio base, (ERB), para o terminal móvel) e 153,6 kbps no enlace reverso (terminal móvel para ERB) fazendo uso de uma portadora e de largura de banda de 1,25 MHz. Para o suporte de altas taxas de dados várias técnicas foram incluídas no IS-856 para otimizar a vazão do enlace direto [5]. Além disso, o IS-856 mantém a compatibilidade com o CDMA2000 e o padrão IS-95.

O outro padrão 3G disponível trata-se do padrão IS-2000, o CDMA 1xEV-DV, definido pelas revisões C e D. Nesta tecnologia 3G a taxa de transmissão de dados por pacote chega ao pico de até 3,0912 Mbps no enlace direto, e 1,8456 Mbps no enlace reverso. Este padrão também garante a compatibilidade com o CDMA2000 e com o padrão IS-95.

Seguindo a linha do GSM vem sendo padronizado pelo 3GPP (*3rd Generation Partnership Project*) o WCDMA conhecido na Europa como UMTS (*Universal Mobile Telecommunications System*). Este padrão define taxas de dados de até 2 Mbps no enlace direto, canais de RF (rádio frequência) com 5 MHz de largura de banda e utiliza um novo espectro de frequência o que implicará em mudanças na estrutura do GSM, porém mantendo a compatibilidade e demais interfaces da arquitetura GSM.

A Figura 1.1 ilustra as várias gerações de telefonia celular tendo como destaque as linhas CDMA e GSM.

## Objetivos do Estudo

Entre todas as tecnologias de celulares 3G disponíveis adotou-se como objeto de estudo nesta dissertação o padrão IS-856.

O sistema IS-856 tem sido alvo de intensa pesquisa [11], [12], [13], [14], [15], [16], [17], [18] e [19].

A maioria dos estudos pesquisados na literatura aborda o desempenho da vazão do sistema IS-856. Em [13] é proposta uma metodologia analítica para avaliação da vazão assumindo tráfego em rajadas. Em [11] e [15] avalia-se o desempenho da vazão através de simulações e exaustivos testes de campo.

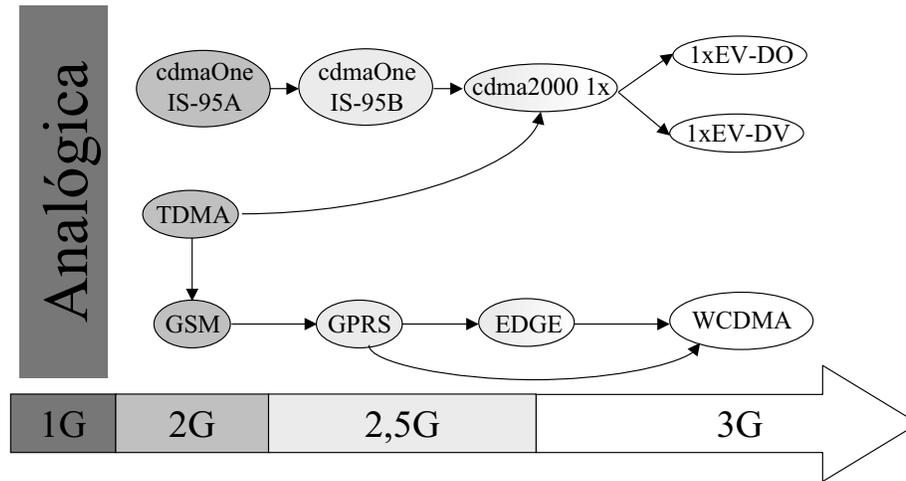


Fig. 1.1: Evolução dos padrões de telefonia celular

Em [14] e [16] são apresentados algoritmos para garantir QoS e fornecer prioridades aos usuários do sistema IS-856. Em [12] são avaliados o atraso de espera dos pacotes, utilização do enlace e vazão de saída usando fontes poissonianas.

Neste trabalho, é feita a avaliação de desempenho do enlace direto do sistema IS-856, considerando modelos reais de tráfego e diversos tipos de escalonamentos. Os modelos reais de tráfegos HTTP (*Hypertext Transfer Protocol*), FTP (*File Transfer Protocol*) e WAP (*Wireless Application Protocol*) propostos em [20] são usados neste estudo com o objetivo de obter medidas de desempenho mais realistas do sistema IS-856. Além disso, são utilizados os escalonadores de pacotes sem prioridade, com prioridade, WFQ (*Weighted Fair Queuing*) e DRR (*Deficit Round Robin*) a fim de verificar a possibilidade de melhoria e garantia de QoS uma vez que os três últimos escalonadores não são definidos pelo padrão IS-856 [10]. A vazão do sistema, o atraso médio dos pacotes, percentual de perda de cada tipo de tráfego e a ocupação do enlace são estudados em função da carga de tráfego. O estudo é feito através de simulação, utilizando a ferramenta de *software* Matlab, e modelo teórico.

No capítulo 2 é apresentada uma visão geral do sistema IS-856 com ênfase na arquitetura do sistema, protocolos de redes utilizados, estrutura do enlace direto e a estrutura do enlace reverso. Além destas características, outras também são descritas de forma sucinta como o uso do escalonador de equidade proporcional, o esquema de *soft handoff* virtual e a característica de redundância incremental.

Os modelos reais de tráfego HTTP, FTP e WAP, os escalonadores de dados sem prioridade ou FIFO (*First In First Out*), com prioridade, WFQ e DRR, a modelagem do sistema IS-856 e os cenários propostos neste trabalho são descritos em detalhes no capítulo 3.

No capítulo 4 são apresentados e analisados os resultados obtidos por meio de simulações dos diversos cenários propostos. Além destas análises também é apresentado um modelo teórico para o sistema IS-856 em que são comparados os resultados das simulações do cenário 4 com os resultados teóricos deste modelo.

Por fim, no capítulo 5 são apresentadas as principais conclusões deste trabalho e são feitos alguns comentários sobre possíveis trabalhos a serem realizados.

# Capítulo 2

## O Sistema IS-856 (1xEV-DO)

### 2.1 Introdução

O sistema IS-856 foi padronizado pelo grupo 3GPP2 (*Third Generation Partnership Project 2*) como uma solução para prover serviços de dados de alta velocidade a redes sem fio móvel e fixa. Em razão das características assimétricas do tráfego IP, o enlace direto torna-se o mais importante dos enlaces [21]. Assim, várias técnicas foram incluídas no IS-856 para otimizar a vazão do enlace direto.

O enlace direto do IS-856 faz uso de avançadas técnicas de modulação e codificação adaptativas, altas taxas de modulação, redundância incremental, diversidade multiusuário, *soft-handoff* virtual, códigos turbo, controle de erro adaptativo e a tecnologia de comutação de pacotes. Em virtude destas características os sistemas 3G permitem realizar entre outros serviços: o acesso rápido à internet e e-mails, vídeo e áudio sob demanda e vídeoconferência.

Neste capítulo para uma melhor compreensão do sistema IS-856 são apresentadas em detalhes algumas das técnicas acima mencionadas com a finalidade de esclarecer os principais conceitos empregados ao longo da dissertação.

### 2.2 Arquitetura da Rede

Como parte da família CDMA2000, o sistema IS-856 é implementado com base nas arquiteturas existentes dos sistemas IS-2000. Entretanto, o sistema IS-856 não utiliza o elemento MSC (*Mobile Switching Center*) presente na arquitetura anterior por já ser uma tecnologia voltada inteiramente para a comutação de pacotes dispensando a estrutura de circuitos empregadas pelas redes de voz.

O padrão IS-856 é um sistema sem fio de ampla área, e sua arquitetura é caracterizada pelos seguintes elementos:

- Dispositivos sem fio: computadores, celulares e dispositivos de mão de uma forma geral que agregam em um mesmo aparelho facilidades como telefone, computador, GPS (*Global Positioning System*), câmera digital, rádio, MP3 *player* etc...
- Rede de acesso: estações-rádio-base ERB em inglês BTS (*Basestation Transceiver Subsystem*), PCF (*Packet Control Function*) normalmente integradas as BSCs(*Base Station Controller*) e controladoras de estações-rádio-base BSC;
- Serviços de rede: PDSN (*Packet Data Serving Node*) que interliga a rede sem fio a internet.

A Figura 2.1 exibe a arquitetura de rede do sistema IS-856.

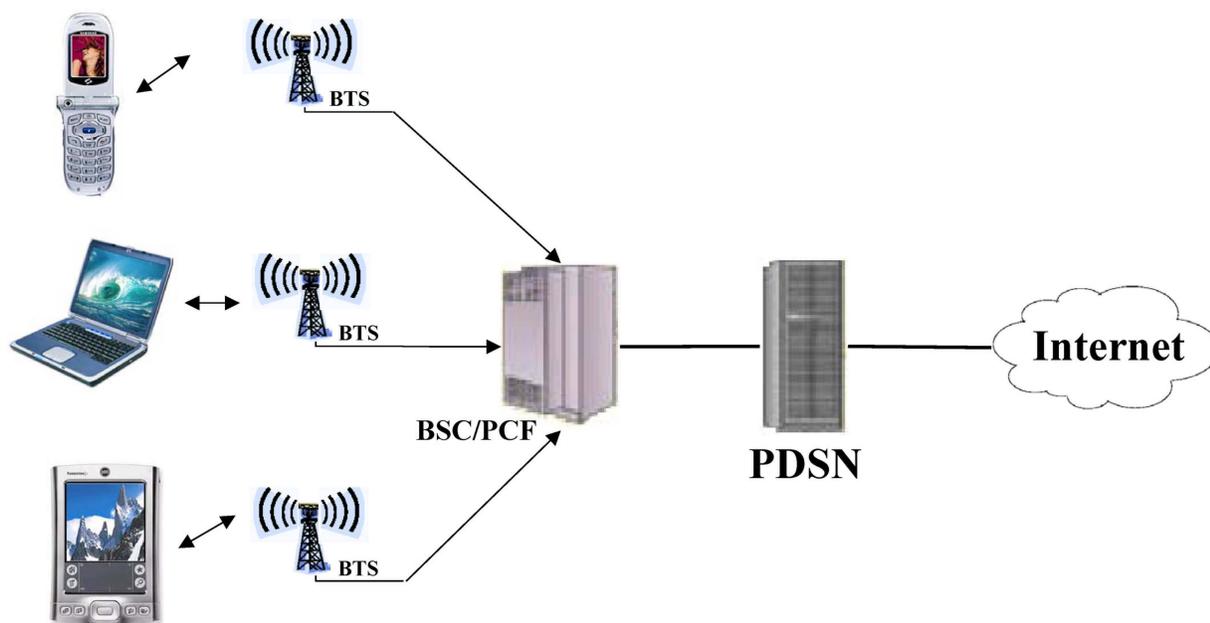


Fig. 2.1: Arquitetura de Rede IS-856

Quando o usuário envia dados, os pacotes de dados dos dispositivos sem fio são enviados pela interface aérea à ERB. Uma vez que os pacotes chegam a ERB estes são encaminhados à BSC. A BSC encaminha os dados a PCF, que encaminha a PDSN. Este caminho também é conhecido como enlace reverso. No enlace direto, ocorre o inverso, os pacotes de dados chegam de uma rede IP e são encaminhados ao PDSN que por sua vez encaminha os pacotes para BSC/PCF. Neste ponto, os pacotes mais uma vez são encaminhados para a ERB e finalmente chegam ao usuário através da interface aérea.

Um grupo de até 10 ERBs são conectadas a uma BSC. A BSC gerencia os *handoffs* mantendo as sessões de dados entre os usuários e controla as conexões dos usuários [22].

O elemento PCF na maioria das vezes é integrado à BSC. Este elemento é responsável pelo gerenciamento de interfaces entre o PDSN e BSC, seleção de PDSN, acompanha os temporizadores de inatividade de usuários e supre esta informação à BSC.

Já PDSN tem como funcionalidade básica o NAS (*Network Access Server*) para estabelecer as sessões de usuários com o protocolo PPP (*Point-to-Point Protocol*). Ambos os serviços IS-2000 e IS-856 utilizam o mesmo PDSN na rede.

## 2.3 Protocolos de Rede

Os sistemas de telecomunicações caminham para a convergência em uma única rede IP que oferece uma vasta gama de serviços conhecida como estrutura *All-IP*. Desta forma, o sistema IS-856 para manter a compatibilidade com aplicações e protocolos baseados em IP utiliza como protocolo de rede o protocolo IP [23]. A Figura 2.2 mostra a pilha de protocolos usados no sistema IS-856.

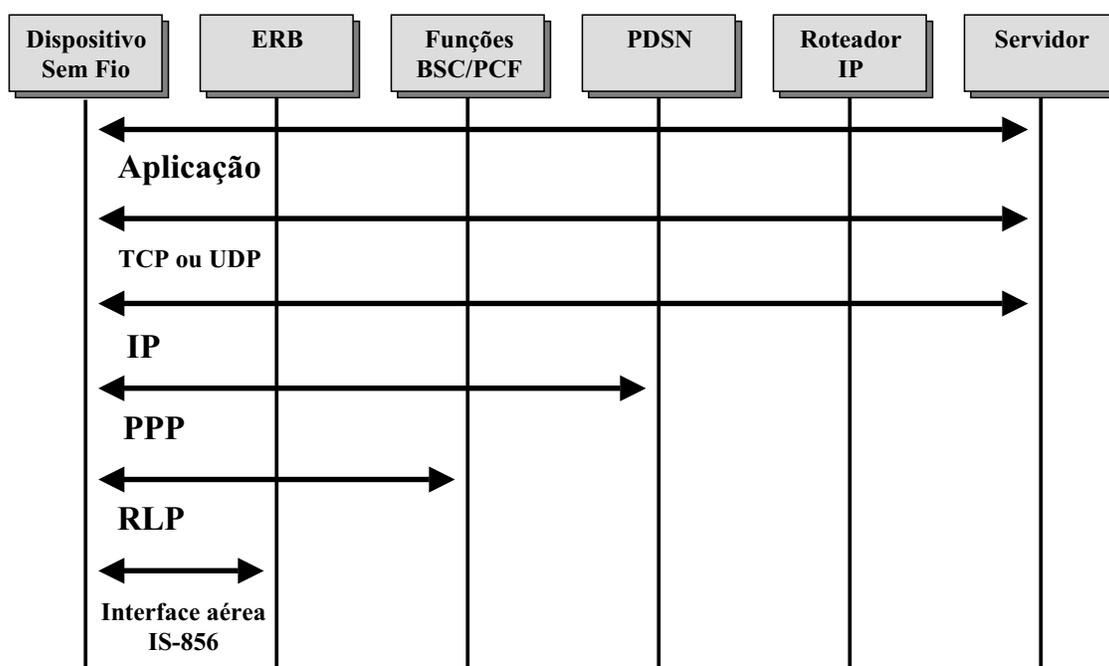


Fig. 2.2: Protocolos de Rede IS-856

A seguir são descritas de forma sucinta as principais características da pilha de protocolos do sistema IS-856.

Acima da camada física do sistema IS-856 está a camada de enlace responsável pela detecção de erros e correção de erros da interface aérea do sistema IS-856. Nesta camada opera o protocolo RLP (*Radio Link Protocol*) que reduz a taxa de erros a um nível onde os protocolos das camadas superiores

podem operar efetivamente. O RLP visa a obtenção de uma taxa de erros inferior a  $10^{-6}$ . Além disso, o RLP também faz seqüenciamento de pacotes.

O protocolo PPP (*Point-to-Point Protocol*) presente na camada de enlace age como um intermediário entre o dispositivo sem fio e o PDSN negociando parâmetros de conexão, compressão do cabeçalho TCP/IP (*Transmission Control Protocol/Internet Protocol*) e o endereço IP a ser alocado. Além disso, inclui o código de redundância cíclica CRC (*Cyclic Redundancy Check*) para identificar os erros na transmissão.

O protocolo IP é um protocolo da camada de rede não orientado à conexão o que implica na independência entre os pacotes, ou seja, os pacotes IP trafegam de forma independente na rede. Em razão desta independência os pacotes podem chegar fora de ordem no destino sendo reordenados pelas camadas superiores. Os dispositivos sem fio podem usar endereços IP estáticos ou endereços IP dinâmicos atribuídos no início de cada sessão.

Na camada de transporte situam-se os protocolos UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*). O protocolo UDP, assim como o protocolo IP, é um protocolo não orientado à conexão regido pela política do melhor esforço sendo aplicado a situações em que não se exige confiabilidade. Por outro lado, o protocolo orientado à conexões TCP garante a confiabilidade do sistema por meio de estabelecimentos de conexões entre dois pontos. Cada conexão é identificada pelo par de endereços IP juntamente com as portas TCP. Além disso, o protocolo TCP provê mecanismos de correção e recuperação de erros.

Para tornar mais fácil o entendimento do funcionamento desta pilha de protocolos é dado o exemplo a seguir proposto em [23] supondo a BSC integrada a PCF. Os pacotes de alguma aplicação da internet chegam ao PDSN com destino a um determinado dispositivo sem fio identificado por um endereço IP levando-se em consideração que cada dispositivo sem fio possui um endereço IP. O PDSN encapsula o pacote IP dentro do cabeçalho PPP. Na seqüência, o PDSN roteia o pacote através do protocolo UDP para o PCF. O PCF remove o cabeçalho UDP, lê a informação do cabeçalho PPP e verifica se o dispositivo está no modo ativo ou inativo; se ativo, encaminha os pacotes para o BSC. O BSC lê o cabeçalho PPP, define em qual ERB e setor localiza-se o dispositivo sem fio, encapsula o pacote PPP em RLP que por sua vez é encapsulado em um cabeçalho UDP e encaminha para a ERB com seu respectivo setor desejado. Finalmente, a ERB remove o cabeçalho UDP e lê a informação RLP e encaminha o pacote para o dispositivo sem fio destinado por meio da interface aérea IS-856.

## 2.4 Interface Aérea - Enlace Direto

A interface aérea do sistema IS-856 foi otimizada para serviços de dados sem fio de alta velocidade oferecendo até 7,4 Mbps por célula (3 setores) através de uma portadora CDMA de largura de

banda 1,25 MHz [1].

Um dos aspectos chave do sistema IS-856 é a alocação de um canal dedicado a dados e outro a voz em contraste com os padrões CDMA anteriores em que o uso de um mesmo canal para voz e dados tornava os serviços ineficientes.

Outro aspecto importante do sistema IS-856 é a multiplexação por divisão de tempo TDM (*Time Division Multiplexing*) em que se maximiza a potência da ERB para um único usuário em um dado instante de tempo. Além desta arquitetura de acesso este sistema também utiliza a multiplexação por divisão de códigos CDM (*Code Division Multiplexing*) presente nos sistemas CDMA anteriores. Portanto, pode-se resumir o sistema IS-856 em um sistema que combina as técnicas de acesso CDM e TDM. A Figura 2.3 mostra a potência usada por setor para os padrões IS-95 e IS-856.

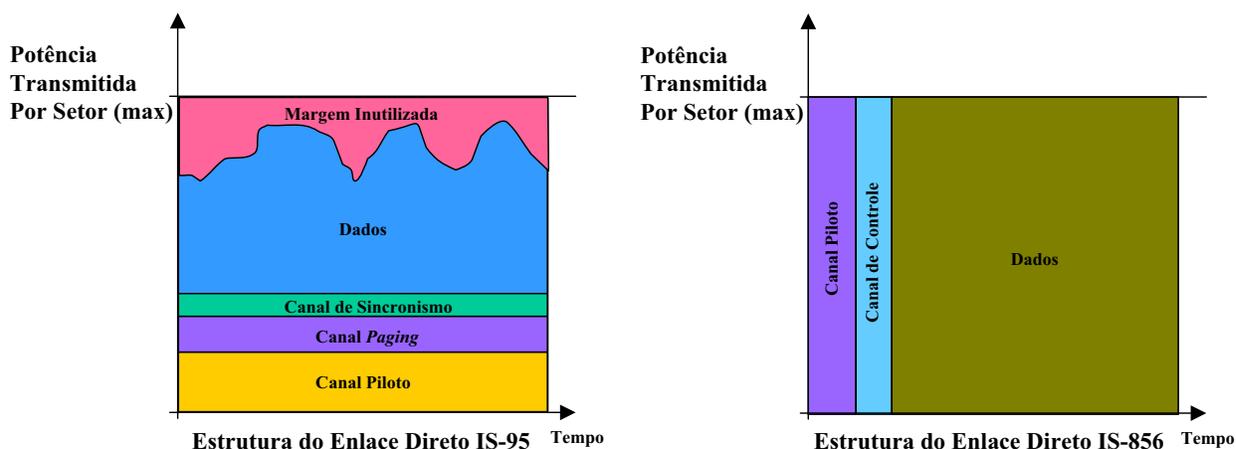


Fig. 2.3: O sistema IS-856 Maximiza a Vazão via Eficiente uso da Potência [1]

Cada canal (Piloto, Sincronismo, *Paging* e Tráfego) no sistema IS-95 é transmitido durante todo o tempo, pois, neste sistema não é utilizada a técnica de compartilhamento de recursos no tempo conhecida como TDM. Além disso, a cada um dos canais é reservada um certa fração do total de potência do setor. Por outro lado, nos canais equivalentes no sistema IS-856 são transmitidos em um certo intervalo de tempo porém, com toda a potência. O uso eficiente dos recursos de potência permite não somente aumentar a área de cobertura da célula, mas também aumentar a relação sinal ruído SINR (*Signal-to-Interference-and-Noise-Ratio*) para um número limitado de usuários [5].

### 2.4.1 Formato do Quadro

O enlace direto do sistema IS-856 é definido em termos de quadros de comprimento de 26,67 ms. Cada quadro é composto por 16 TS (*Time-Slots*) com duração de 1,67 ms, como mostra a Figura 2.4.

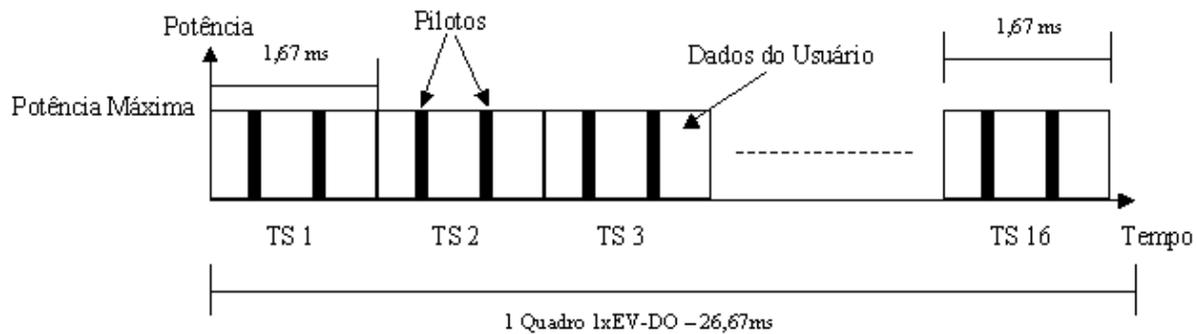


Fig. 2.4: Protocolos de Rede IS-856

A condição do canal é determinada pela transmissão de pilotos no enlace direto e o terminal informa a condição do canal através do enlace reverso. Os pilotos são transmitidos em intervalos pré-determinados dentro de cada TS como pode ser visto na Figura 2.4.

## 2.4.2 Taxas de Transmissão

O enlace direto do sistema IS-856 oferece uma gama de taxas de dados podendo atingir até 2,457 Mbps. Entretanto, sob as condições de um ambiente móvel sem fio, o sinal de rádio frequência pode variar significativamente. Desta forma, de acordo com a potência do sinal recebido o terminal pode suportar esquemas de modulação de ordem superior (mais bits por símbolo) e códigos de erro mais fracos ou utilizar esquemas de modulação mais robustos e códigos de erros mais fortes com taxas de dados menores.

As taxas de dados suportadas pelo enlace direto variam de 38,4 kbps a 2457,6 kbps por setor de uma célula. Um dos três esquemas de modulação QPSK (*Quadrature Phase Shift Keying*), 8PSK (*8 Phase Shift Keying*) e 16QAM (*16-positions Quadrature Amplitude Modulation*) é utilizado dependendo da taxa de dados. Além disso, diferentes quantidades de TSs são alocados para cada taxa de dados. A Tabela 2.1 exhibe em detalhes as diferentes taxas de dados em função da modulação, TS, o tamanho dos pacotes, taxa de código, fator de repetição e o tempo de duração dos pacotes em ms.

A quantidade de TSs alocados para cada taxa de dados varia de acordo com um critério de equidade, pois, seria injusto alocar mais TSs para taxas maiores e menos TSs para taxas menores. Desta forma, para manter um desempenho equilibrado entre as vazões de todos os usuários ativos sob diversas condições de canais são alocados mais TSs para taxas menores e menos TSs para taxas maiores. Por exemplo, na Tabela 2.1 são alocados 16 TSs para a taxa de 38,4 kbps e 1 TS para a taxa de 2457,6 kbps.

Outra consequência do critério de equidade é a escolha dos tamanhos dos pacotes entre 1024 bits

Tab. 2.1: Tipos de Modulação por Taxas de Dados - Enlace Direto

Taxa (kbps)	Bits por Pacote	Time Slots	Modulação	Taxa de Código	Fator de Repetição	Duração/Pacote (ms)
<b>38,4</b>	1024	16	QPSK	1/5	9,6	26,67
<b>76,8</b>	1024	8	QPSK	1/5	4,8	13,33
<b>153,6</b>	1024	4	QPSK	1/5	2,4	6,67
<b>307,2</b>	1024	2	QPSK	1/5	1,2	3,33
<b>614,4</b>	1024	1	QPSK	1/3	1	1,67
<b>921,6</b>	3072	2	8PSK	1/3	1	3,33
<b>1228,8</b>	2048	1	QPSK	1/3	1	1,67
<b>1843,2</b>	3072	1	8PSK	1/3	1	1,67
<b>2457,6</b>	4096	1	16QAM	1/3	1	1,67

e 4096 bits. Os tamanhos dos pacotes necessitam ser grandes o suficiente para explorar o ganho oferecido pelas taxas de código, enquanto minimiza o impacto do encapsulamento de dados nas camadas superiores (*overhead*). Por outro lado, os tamanhos dos pacotes precisam ser pequenos o suficiente para evitar o atraso excessivo e o empacotamento ineficiente dos dados pequenos. Dessa forma, o tamanho dos pacotes adotados neste padrão restringiu-se a pacotes de 1024 bits, 2048 bits, 3072 bits e 4096 bits conforme a Tabela 2.1.

O mecanismo corretor de erro adotado no sistema IS-856 é o código convolucional concatenado do tipo turbo (*turbo code*) em que são introduzidas redundâncias na informação em vários níveis de complexidade [4]. Os dados do usuário são codificados no enlace direto a uma taxa de 1/5 ou 1/3 de acordo com os tamanhos dos pacotes. O codificador turbo é constituído de dois codificadores convolucionais em paralelo com um *interleaver* responsável pelo embaralhamento de informações para que o erro ocorrido em uma seqüência de bits não atrapalhe a comunicação com o receptor das informações do usuário. Na saída deste codificador turbo os dados codificados são concatenados e repetidos até atingir o nível desejado de símbolos dependendo do fator de repetição exibido na Tabela 2.1. Maiores informações sobre o codificador turbo podem ser obtidas em [10] e [24].

## 2.5 Escalonador de Equidade Proporcional

O padrão IS-856 deixa em aberto o escalonador de dados a ser usado no sistema. Dessa maneira, muitos fabricantes incluíram seus próprios algoritmos de escalonamento. Um dos escalonamentos propostos pela indústria é o chamado escalonador de equidade proporcional (*Proportional Fair Scheduling*) em que incorpora duas importantes características: diversidade multiusuário e maximização

do critério de equidade.

Este escalonamento baseia-se no conceito de servir o usuário quando este estiver com uma condição do sinal acima da média observada, ou seja, são atribuídas "prioridades" aos usuários que estão com a potência do sinal acima da média e que transmitiram há mais tempo. No entanto, todos os usuários são tratados pelo algoritmo de mesma forma. Assim, este escalonamento aumenta a probabilidade de servir o usuário com altas taxas de dados tomando como vantagem a variabilidade do canal.

Neste trabalho o escalonador de equidade proporcional não foi abordado em razão da alta complexidade agregada a ele dificultando a implementação do mesmo. Além disso, este algoritmo não garante a QoS necessária para atender as diferentes necessidades de cada usuário [14]. Entretanto, como o padrão IS-856 não especifica escalonadores de dados foram estudados nesta dissertação outros escalonadores amplamente usados na literatura para outros fins. Entre os escalonadores existentes foram estudados os escalonamentos FIFO (*First In First Out*), com prioridade, WFQ (*Weighted Fair Queuing*) e DRR (*Deficit Round Robin*) a fim verificar as possibilidades de melhoria e garantias de QoS. Estes escalonamentos são apresentados em detalhes no Capítulo 3.

## 2.6 Soft Handoff Virtual

A técnica de *handoff* utilizada no sistema IS-856 chama-se *soft handoff virtual*. Nesta técnica ao invés de usar mais de um setor para servir um mesmo terminal a fim de se diminuir a queda de chamadas como acontece nos sistemas IS-95 o sistema IS-856 utiliza um canal lógico chamado de DRC (*Data Rate Control*) na qual leva a informação de qual setor da ERB deve servi-lo, isto é, informa o único setor da ERB que irá transmitir para o terminal. Desta forma, o IS-856 baseia-se na rápida seleção de qual setor irá lhe atender de acordo com a potência do sinal que o terminal observa de cada um dos setores. Este esquema de *handoff* elimina o *overhead* dos sistemas IS-95 que usam *soft handoff*.

## 2.7 Redundância Incremental

A redundância incremental ou *hibrid-ARQ* ou ainda a terminação antecipada dos pacotes é aplicada a pacotes que utilizam mais de um TS sendo transmitidos com um espaçamento de 4 TSs como ilustra a Figura 2.5. O espaçamento de 4 TSs permite ao terminal decodificar parte do pacote e enviar um ACK (*acknolegment*) para a ERB caso tenha conseguido decodificar completamente a informação sem necessitar dos TSs restantes [4] que possui informações redundantes. Desta forma, caso a ERB receba o ACK, a transmissão do pacote é terminada aumentando a taxa de dados sob o

ponto de vista do usuário.

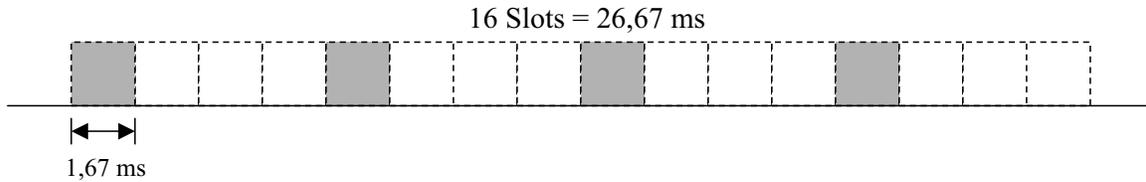


Fig. 2.5: Pacote de 2 TSs com Intervalo de Transmissão de 4 TSs

## 2.8 Interface Aérea - Enlace Reverso

A estrutura do enlace reverso do sistema IS-856 é bastante similar à estrutura presente no sistema IS-2000. Os canais de dados suportam taxas de dados de 9,6 kbps até 153,6 kbps com 16 TSs por pacote. As baixas taxas de transmissões de dados no enlace reverso em comparação ao enlace direto não são um fator limitante do sistema IS-856 em virtude das características assimétricas dos tráfegos de dados das redes fixas ou móveis atualmente. Esta característica está relacionada ao maior volume de tráfego presente no enlace direto do que no enlace reverso.

O enlace reverso, diferentemente do enlace direto, suporta apenas um único esquema de modulação o BPSK (*BiPhase Shift Keying*). Este esquema de modulação aliado a taxa de codificação dos pacotes de 1/2 e 1/4 e ao codificador turbo permite transmitir dados com taxa de até 153,6 kbps. A Tabela 2.2 exibe as taxas de dados do enlace reverso versus o tamanho do pacote, o tipo de modulação, as taxas de código e a duração do empacotamento do pacote em ms. Maiores detalhes sobre o enlace reverso podem ser conferidos em [25].

Tab. 2.2: Tipos de Modulação por Taxas de Dados - Enlace Reverso

Taxa (kbps)	Bits por Pacote	Time Slots	Modulação	Taxa de Código	Duração/Pacote (ms)
9,6	256	16	BPSK	1/4	26,67
19,2	512	16	BPSK	1/4	26,67
38,4	1024	16	BPSK	1/4	26,67
76,8	2048	16	BPSK	1/4	26,67
153,6	4096	16	BPSK	1/2	26,67

## 2.9 Conclusão

O sistema IS-856 apresenta altas taxas de dados no enlace direto do sistema. Estas altas taxas são obtidas face à avançadas técnicas de modulação e codificação adaptativas, altas taxas de modulação, redundância incremental, diversidade multi-usuário, *soft-handoff* virtual, codificadores do tipo turbo, controle de erro de pacotes adaptativo e a tecnologia de comutação de pacotes. Entretanto, o desempenho do sistema dependerá da variabilidade da condição do canal e do número de usuários ativos. No próximo capítulo são apresentados a modelagem do sistema IS-856 considerando diversas fontes reais de tráfego e diferentes tipos de escalonadores.

# Capítulo 3

## Modelos Reais de Tráfego, Escalonadores e Modelagem do Sistema IS-856

### 3.1 Introdução

A maioria dos estudos presentes na literatura avalia o desempenho do sistema IS-856 através de fontes poissonianas. Neste capítulo são abordados os modelos reais de tráfego HTTP, FTP, WAP propostos em [20] a fim de se obter um desempenho mais realista do sistema IS-856. Para todas as fontes considerou-se que os usuários HTTP, WAP e FTP estão sempre ativos, isto é, fazem as requisições, aguardam a resposta, esperam o tempo de leitura e então fazem a próxima requisição.

Além disso, neste capítulo são apresentados os escalonadores de dados adotados neste estudo sendo estes: FIFO (*First In First Out*), com prioridade, WFQ (*Weighted Fair Queuing*) e DRR (*Deficit Round Robin*).

Por fim, com a finalidade de se avaliar o comportamento do sistema IS-856 face a diversos tipos de escalonamentos e diferentes modelos reais de tráfego, foram simulados diversos cenários com diferentes ênfases. As simulações foram realizadas por meio da ferramenta de *software* Matlab. Desta forma, no final deste capítulo são descritos os parâmetros necessários para as simulações, as suposições adotadas e os cenários de simulações.

### 3.2 Modelo TCP

O protocolo TCP (*Transmission Control Protocol*) ou UDP é utilizado como protocolo de transporte nas redes IP. Desta forma, para aumentar a precisão dos modelos de tráfego HTTP e FTP foi introduzido o modelo TCP como base dos modelos de tráfego HTTP e FTP.

O modelo TCP apresentado neste trabalho é uma versão simplificada do proposto em [20]. Nesta

versão é descartado o mecanismo de controle de congestionamento conhecido como partida lenta. Desta forma, após o estabelecimento da conexão entre o terminal móvel e a ERB são transmitidos os segmentos de dados TCP. Na seqüência ao término desta transmissão a conexão é encerrada como ilustra a Figura 3.1.

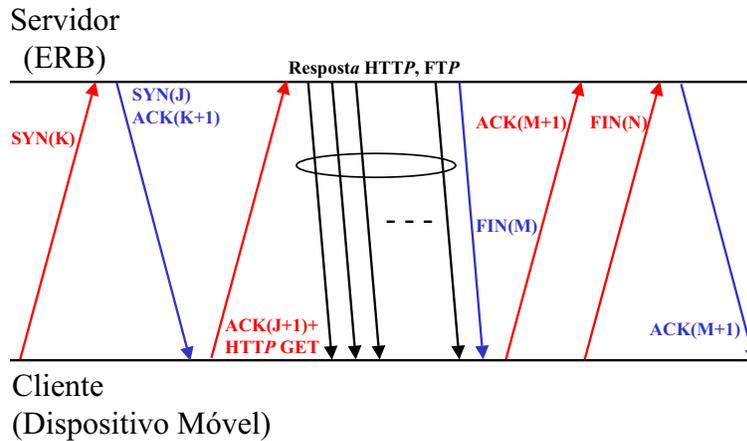


Fig. 3.1: Segmentos de Controle em uma Conexão TCP

O tempo total gasto pelos segmentos TCP no sistema IS-856 é dado pela seguinte expressão:

$$\tau_{total} = \tau_c + \tau_1$$

Sendo:

$\tau_c$  representa a soma dos seguintes tempos: o tempo levado por um pacote ACK propagar do cliente para o servidor, mais o tempo de processamento deste pacote pelo servidor e o tempo gasto pelo segmento de dado TCP propagar do servidor para a ERB.

$\tau_1$  o tempo de transmissão de um segmento de dado TCP da ERB para o dispositivo móvel por meio da interface aérea.

Neste modelo TCP,  $\tau_c$  é modelado por uma variável aleatória distribuída exponencialmente com média de 50 ms enquanto  $\tau_1$  é modelado por uma variável determinística determinada dinamicamente pelas taxas de transmissões de dados presentes na Tabela 2.1.

A Figura 3.2 mostra a interação entre cliente, ERB e servidor através dos meios rede núcleo e interface aérea. Pode ser observado nesta figura o estabelecimento de uma conexão TCP entre cliente e servidor, a transmissão dos segmentos TCP do servidor para o cliente e por fim o fechamento da conexão TCP.

Com base nesta observação, o processo de chegada de pacotes na ERB é exibido pela Figura 3.3 e também descrito no algoritmo de passos abaixo.

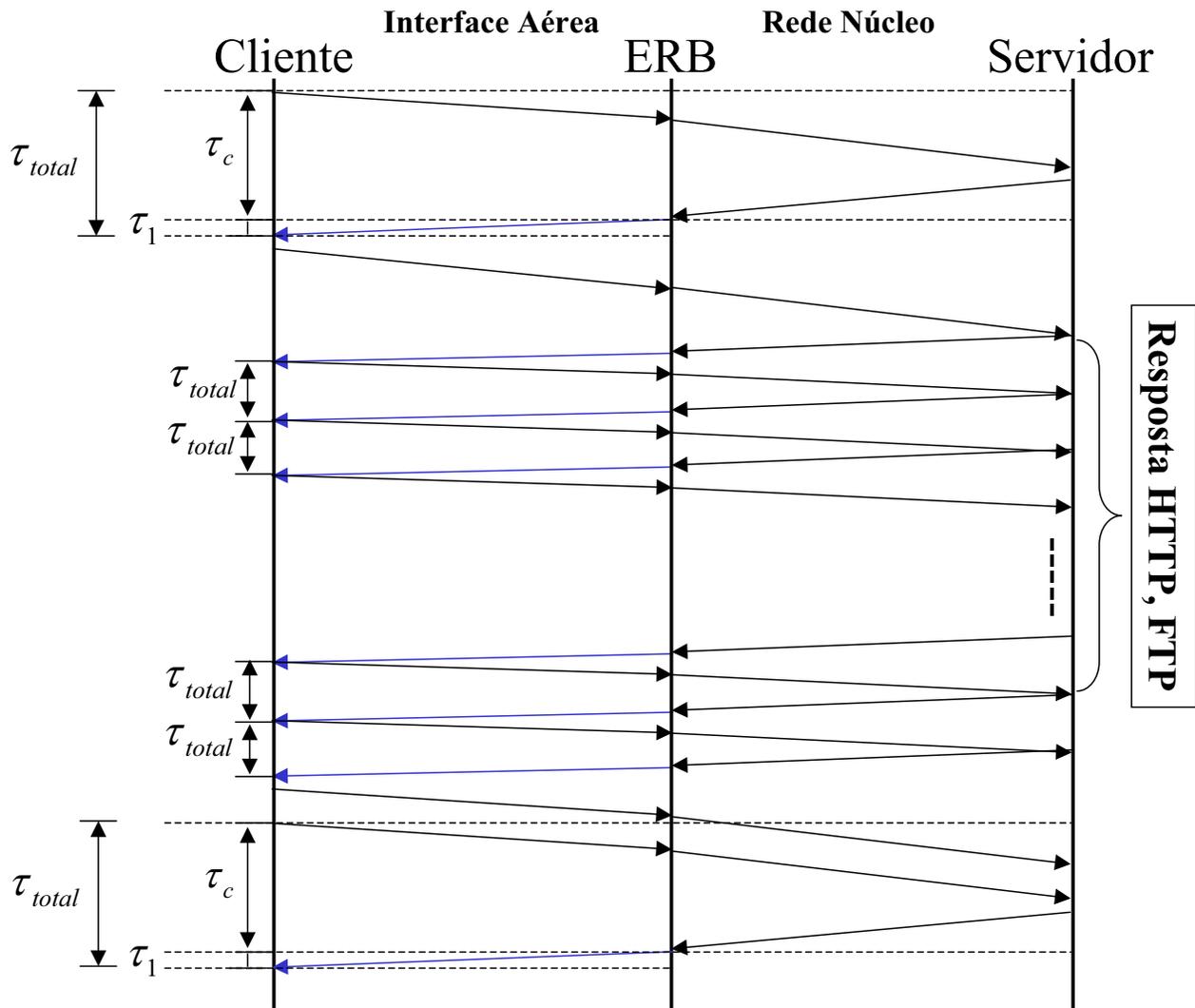


Fig. 3.2: Controle de Fluxo TCP

1. O tamanho do objeto em bytes é armazenado na variável  $S$ . Calcula-se o número de pacotes do objeto,  $N = \lceil S / (MTU - 40) \rceil$ .
2. Todos os pacotes a serem transmitidos são enfileirados numa fila de acordo com a ordem de chegada dos pacotes. Se  $P == 0$ , vá para o passo 6.
3. Aguarda a transmissão de um pacote do objeto sobre a interface aérea.
4. Escalona a chegada do próximo pacote do objeto após um intervalo de tempo  $\tau_c$ . Se  $P == 1$ , então  $P = 0$ , senão  $P = P - 1$ .

5. Se  $P > 0$  vá para o passo 3.

6. Fim

Os segmentos de dados TCP possuem unidade máxima de transferência MTU (*Maximum Transmission Unit*) de 1500 bytes ou de 576 bytes. A estrutura deste segmento destina 1460 bytes ou 536 bytes para dados considerado segmentos com MTU de 1500 bytes e 576 bytes respectivamente, identificados em [20] de MTU-40. Os outros 40 bytes de cada MTU são dedicados a códigos de redundância e cabeçalhos TCP e PPP em que já está encapsulado o cabeçalho IP. Desta forma, os segmentos de dados TCP possuem um *overhead* percentual de 2,7% para segmentos com MTU de 1500 bytes e 7% para segmentos com MTU de 576 bytes.

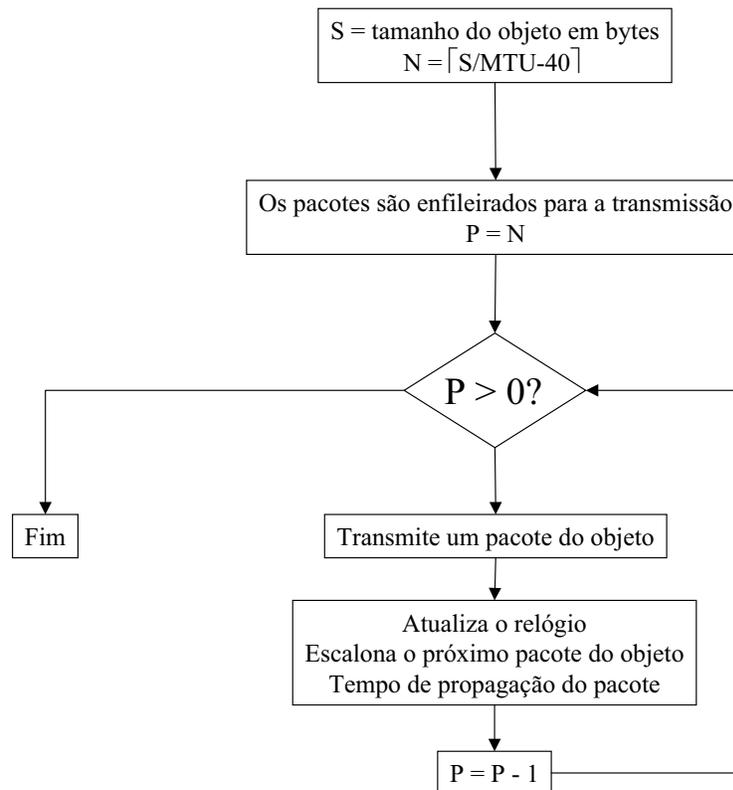


Fig. 3.3: Processo de Chegada de Pacotes na ERB

### 3.3 Modelo HTTP

#### 3.3.1 Características do Modelo de Tráfego HTTP

A Figura 3.4 ilustra uma típica seção de tráfego web. Esta seção apresenta períodos de *downloads* de páginas web e períodos de tempo de leitura destas páginas web sendo que em uma seção pode haver  $n$  períodos de *downloads* e leituras de páginas web. Esta divisão em períodos do tráfego HTTP deve-se a interação homem máquina. Além disso, outra característica importante do tráfego web é a autosimilaridade, isto é, estatísticas similares em diferentes escalas de tempo [20].

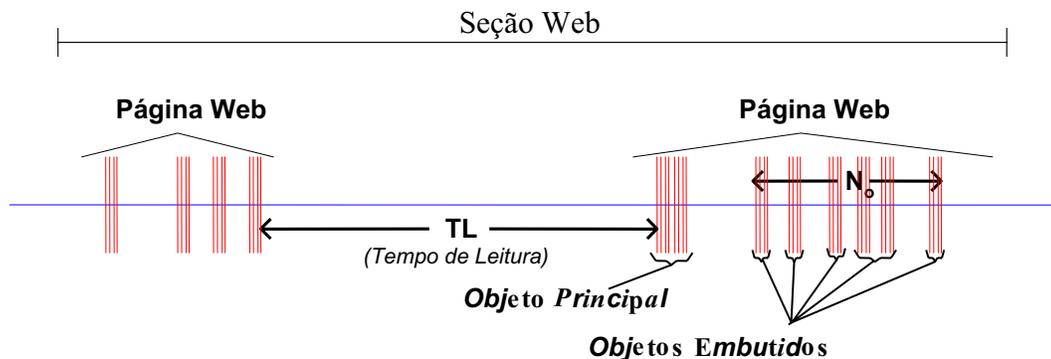


Fig. 3.4: Seção de Navegação Web

Uma página web é composta por um único objeto principal e vários objetos embutidos. Inicialmente, o navegador busca o objeto principal da página HTML (*Hyper Text Markup Language*) usando uma requisição HTTP GET. Após receber o objeto principal, o navegador determina o *layout* da página para introduzir objetos embutidos tais como figuras, botões estilizados entre outros objetos. O tempo gasto para a determinação do *layout* da página web é chamado tempo *parsing*. Em seguida, o navegador baixa os objetos embutidos e o usuário inicia a leitura desta página web. A busca e o *download* dos objetos principal e embutidos são representados pelos períodos *On* enquanto o tempo de leitura destas páginas representam o período *Off*. A Figura 3.5 mostra o objeto principal e seus respectivos objetos embutidos da página web do jornal *Wall Street*.

As características do tráfego HTTP dependerão do modo de *download* dos segmentos TCP usadas pelos servidores web e navegadores. Atualmente duas versões do protocolo HTTP são usadas, HTTP/1.0 e HTTP/1.1, na maioria dos servidores e navegadores. Estas duas versões diferem na maneira como são usadas as conexões TCP para transferir os objetos principais e embutidos.

No HTTP/1.0, uma conexão TCP distinta é usada para o *download* de cada objeto principal ou embutido. Esta versão também conhecida como *HTTP/1.0-burst mode transfer* é a mais popular entre os navegadores web. São permitidas até 4 conexões TCP paralelas na maioria dos navegadores.

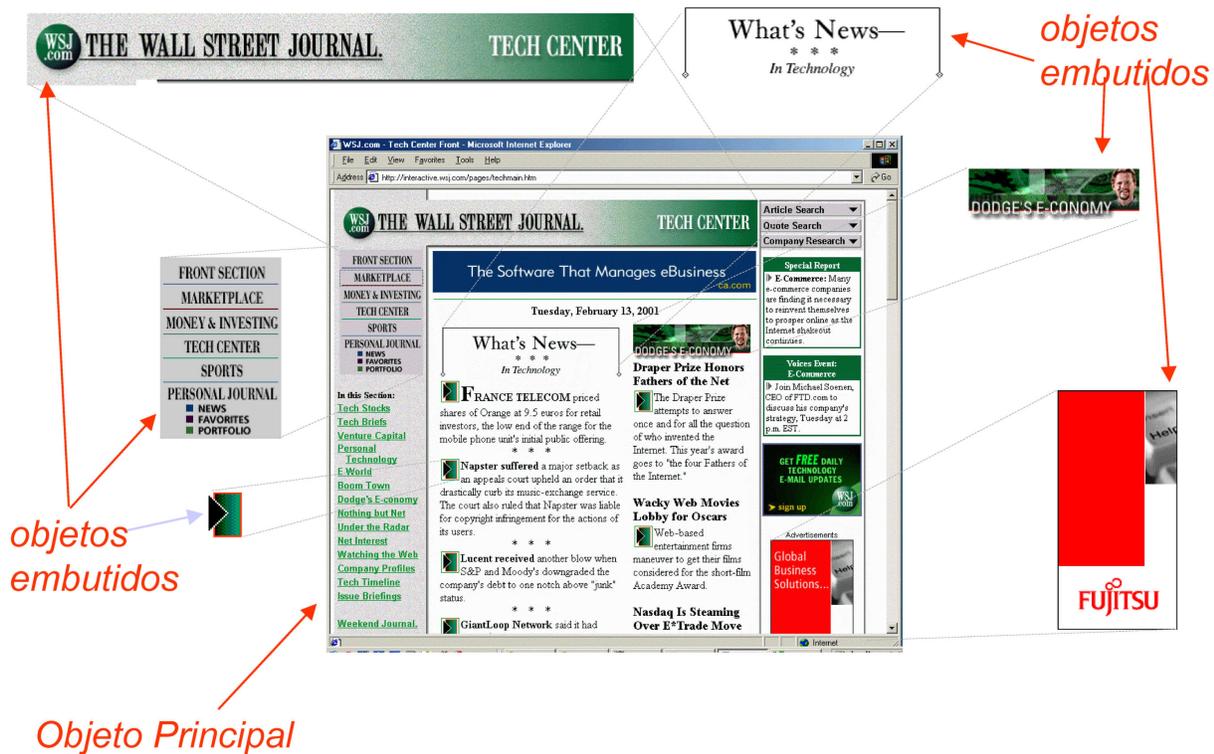


Fig. 3.5: Uma Típica Página Web e seu Conteúdo

Já no HTTP/1.1, também conhecido como *HTTP/1.1-persistent mode transfer* são utilizadas conexões TCP persistentes em que os objetos são baixados serialmente por meio de uma única conexão TCP.

### 3.3.2 Parâmetros do Modelo de Tráfego HTTP

As distribuições dos parâmetros do modelo de tráfego HTTP foram baseadas em uma pesquisa rigorosa na literatura e propostas em [20]. Os parâmetros estão descritos na Tabela 3.1.

Uma página web é representada por um único objeto principal e vários objetos embutidos. O tamanho do objeto principal segue uma distribuição lognormal com o tamanho mínimo de 100 bytes, máximo de 2 megabytes e média de 10710 bytes com os parâmetros  $\sigma$  valendo 1,37 e  $\mu$  sendo 8,35. Já os objetos embutidos possuem distribuição lognormal com tamanho médio de 7758 bytes, mínimo de 50 bytes e máximo de 2 megabytes sendo os parâmetros  $\sigma$  com valor de 2,36 e  $\mu$  valendo 6,17.

O número de objetos embutidos ( $N_o$ ) por página web obedece à distribuição pareto com média de 5,64 objetos e máximo de 53 objetos com os parâmetros  $k$  com valor 2,  $\alpha$  sendo 1,1 e  $m$  valendo 55.

O tempo de leitura (TL) é o tempo entre duas requisições de página web distribuído exponencial-

Tab. 3.1: Parâmetros do Modelo de Tráfego HTTP

Parâmetros		Média	Mínimo	Máximo	Distribuição
Tamanho do Objeto	Principal	10710 bytes	100 bytes	2 Mbytes	Lognormal
	Embutido	7758 bytes	50 bytes	2 Mbytes	Lognormal
Nº de Objetos Embutidos p/pág. ( $N_o$ )		5,64		53	Pareto
Tempo de Leitura (TL)		30 s			Exponencial
Tempo <i>Parsing</i>		0,13			Exponencial

mente com média 30 segundos.

Por fim, o tempo *parsing*, que é o tempo gasto para determinar o *layout* da página web após a busca do objeto principal, obedece também a uma distribuição exponencial com média 0,13 segundos [26].

O modelo TCP adotado neste estudo estabelece o processo de chegada dos pacotes à ERB. Cada objeto ou arquivo dos modelos HTTP e FTP é segmentado em pacotes de 576 bytes e 1500 bytes. Definiu-se uma distribuição de 24% para 576 bytes e 76% para 1500 bytes [20] e outra distribuição de 50% entre os dois modos de *download* disponíveis.

Tab. 3.2: Funções Densidades das Distribuições

Distribuição	PDF	Parâmetros
<b>Lognormal</b>	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x \geq 0$	$\sigma, \mu$
<b>Pareto</b>	$f_x = \frac{\alpha k^\alpha}{x^{\alpha+1}}, k \leq x < m$ $f_x = \left(\frac{k}{m}\right)^\alpha, x = m$	$\alpha, k, m$
<b>Exponencial</b>	$f_x = \lambda \exp(-\lambda x), x \geq 0$	$\lambda$
<b>Geométrica</b>	$f = p(1 - p)^n$	$p, n$

A Tabela 3.2 apresenta todas funções densidades probabilísticas (PDF) usadas para obter os modelos de tráfegos HTTP, FTP e WAP. No entanto, para a geração de números aleatórios com as distribuições lognormal, pareto, exponencial e geométrica foi adotado neste estudo o método da transformação inversa em que as distribuições podem ser invertidas analiticamente com exceção da distribuição lognormal. Este método foi aplicado em todos os modelos de tráfego estudados. Entretanto, para a geração dos números aleatórios com as distribuições citadas acima grande parte já estava implementada no *toolbox stat* da ferramenta de *software* Matlab com exceção da distribuição Pareto obtida em [27].

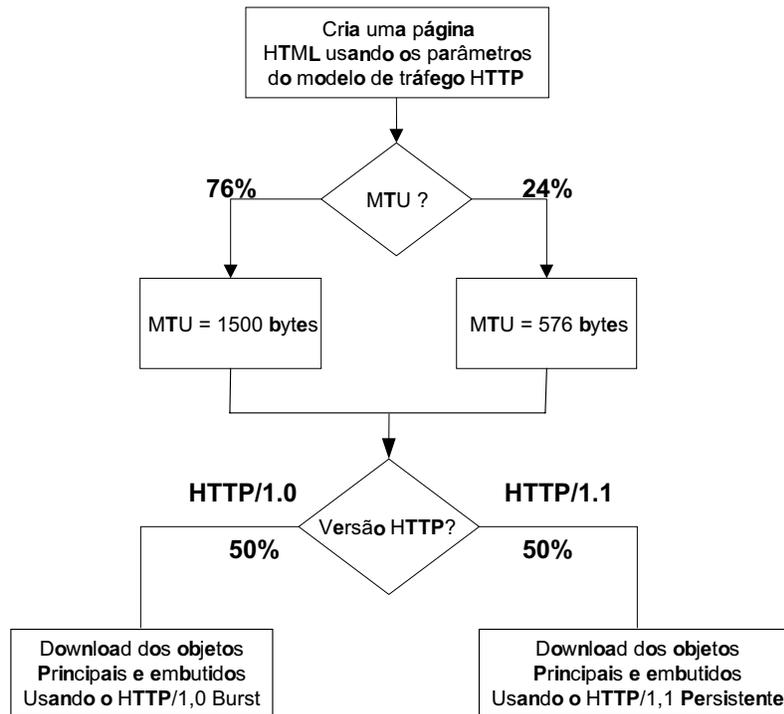


Fig. 3.6: Modelagem de *Download* de uma Página Web

### 3.3.3 Histogramas das Fontes HTTP

A Figura 3.7 mostra o histograma das fontes HTTP. Além dos pacotes de 1500 bytes, 576 bytes há também os pacotes com tamanho de 80 bytes, 160 bytes, 40 bytes, etc que representam os pacotes de controle e sincronismo.

Na Figura 3.8 é exibida a ordem de chegadas dos pacotes do tráfego HTTP. Nesta figura são alternados períodos de rajadas de pacotes com períodos de silêncio.

A Figura 3.9 mostra o histograma da distribuição do intervalo (em tempo) entre chegadas de pacotes do tráfego HTTP. Nesta figura, a maioria dos pacotes concentra-se de torno de 50 ms que representam os períodos de tráfego ativo enquanto os períodos de inativos estão representados por alguns pontos do gráfico próximos ao eixo rotulado Tempo (s).

## 3.4 Modelo FTP

Nas aplicações FTP, a seção FTP consiste de uma seqüência de  $n$  arquivos transferidos, separados pelos tempos de leitura como, por exemplo, ilustra Figura 3.10.

O tamanho do arquivo segue uma distribuição lognormal com tamanho médio de 2 megabytes e

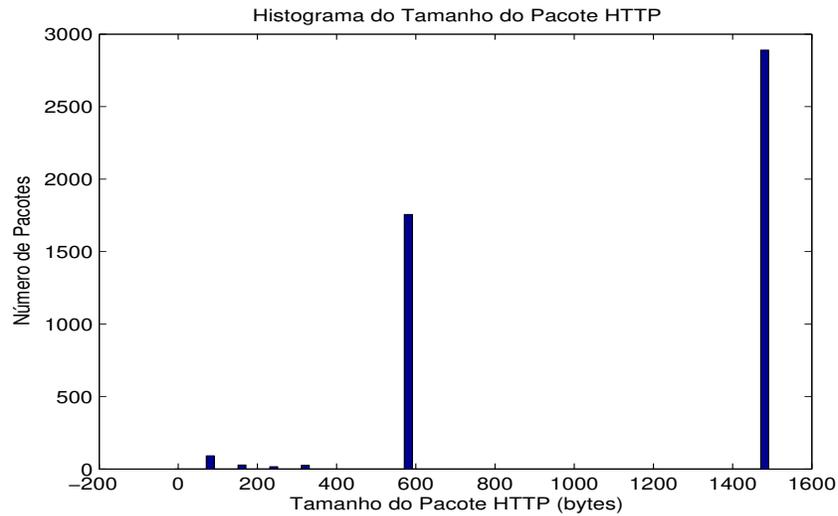


Fig. 3.7: Histograma do Comprimento do Pacote HTTP

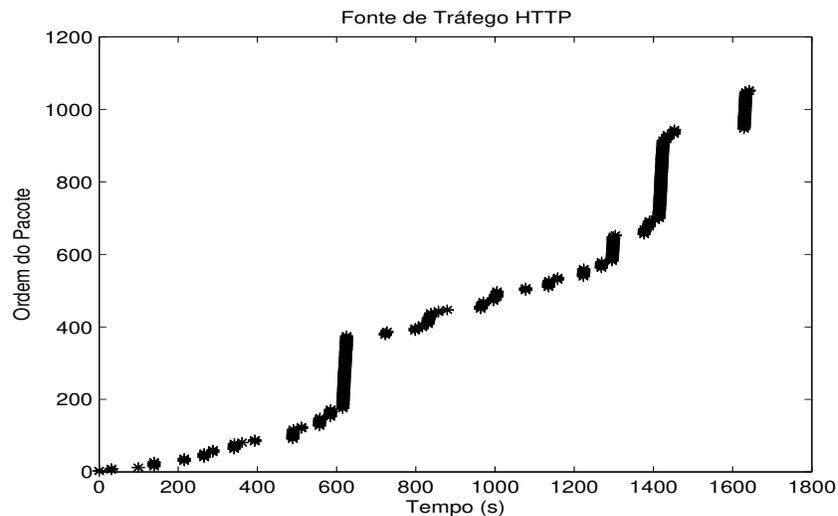


Fig. 3.8: Ordem de Chegadas dos Pacotes do Tráfego HTTP

máximo de 5 megabytes com os parâmetros  $\sigma$  sendo 0,35 e  $\mu$  valendo 14,45.

O tempo de leitura (TL) é o tempo entre *downloads* de diferentes arquivos distribuído exponencialmente com média 180 segundos como mostra a Tabela 3.3.

Com base na distribuição de pacotes mostrada na Seção 3.3.2 os arquivos FTP são transferidos 76% das vezes usando um MTU de 1500 bytes e 24% das vezes utilizando um MTU de 576 bytes. O processo de chegada de pacotes na ERB é descrito pelo fluxograma da Seção 3.2. A seguir na Figura 3.11 é mostrado o processo de geração de tráfego FTP.

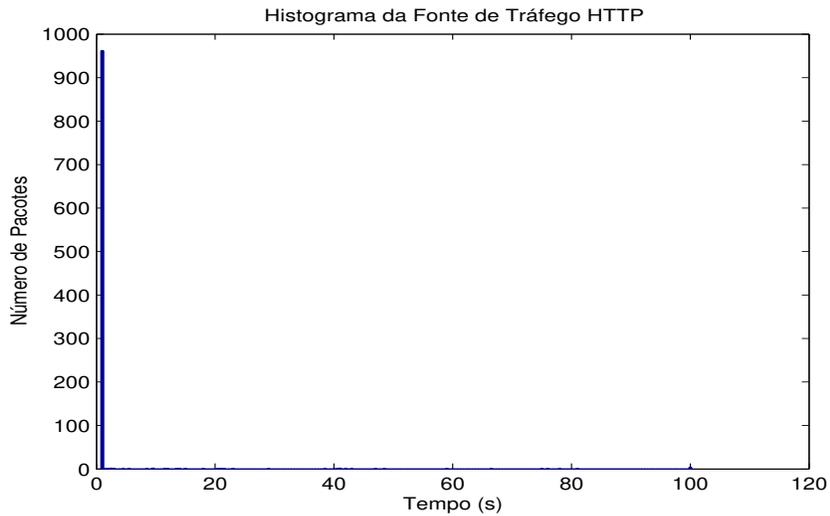


Fig. 3.9: Histograma dos Intervalos entre Chegadas do Tráfego HTTP

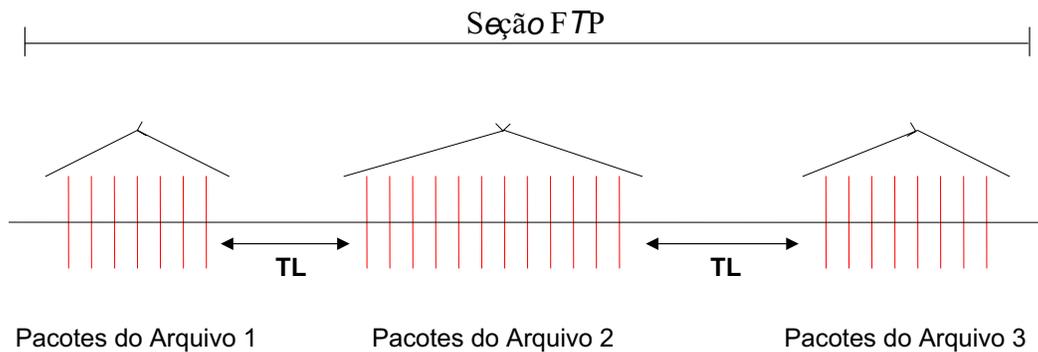


Fig. 3.10: Típica Seção FTP

Tab. 3.3: Parâmetros do Modelo de Tráfego FTP

Parâmetros	Média	Máximo	Distribuição
Tamanho do Arquivo	2 Mbytes	5 Mbytes	Lognormal
Tempo de Leitura	180 s		Exponencial

### 3.4.1 Histogramas das Fontes FTP

A Figura 3.12 mostra o histograma do tamanho dos pacotes das fontes FTP. Neste histograma há apenas pacotes com tamanho de 1500 bytes ou pacotes com 576 bytes.

Na Figura 3.13 é ilustrada a ordem de chegada dos pacotes das fontes FTP. Esta fonte alterna

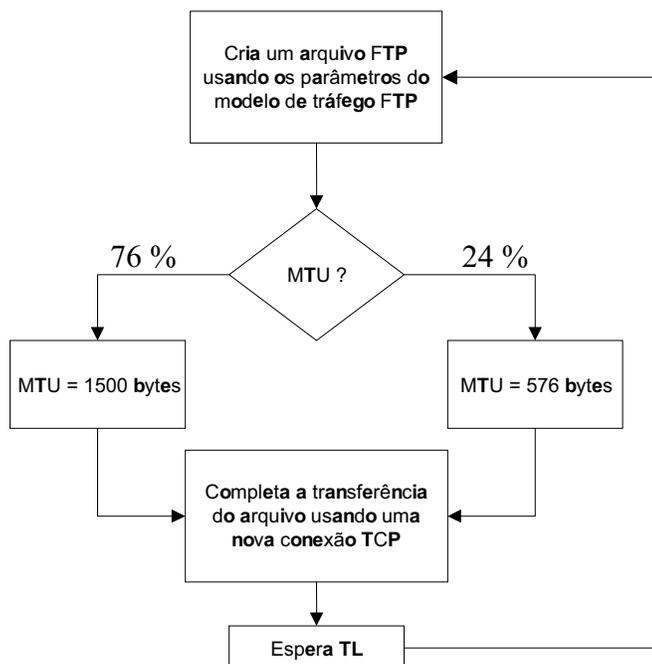


Fig. 3.11: Modelo de Tráfego FTP

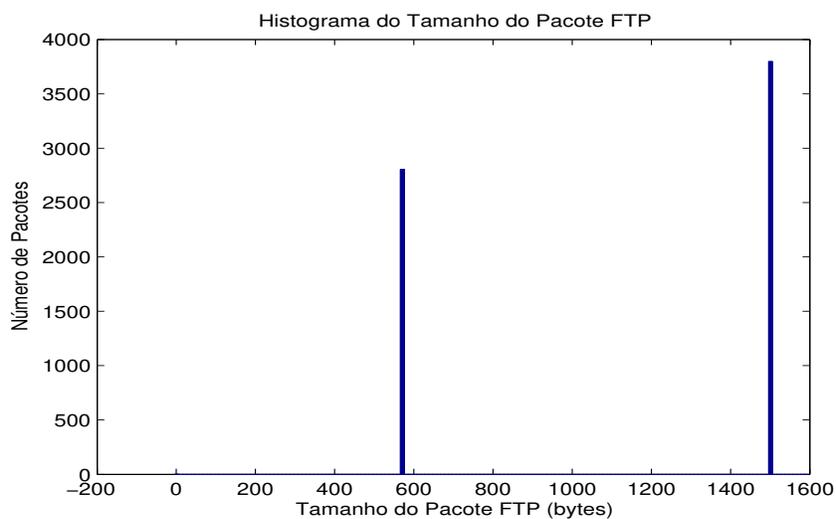


Fig. 3.12: Histograma do Comprimento do Pacote FTP

longos períodos de tráfego intenso com longos períodos inatividade.

A Figura 3.14 exibe o histograma da distribuição do intervalo (em tempo) entre chegadas de pacotes do tráfego FTP. Nesta figura, assim como acontece na Figura 3.9 a maioria dos pacotes concentra-se de torno de 50 ms que representam os períodos de transmissões dos pacotes enquanto

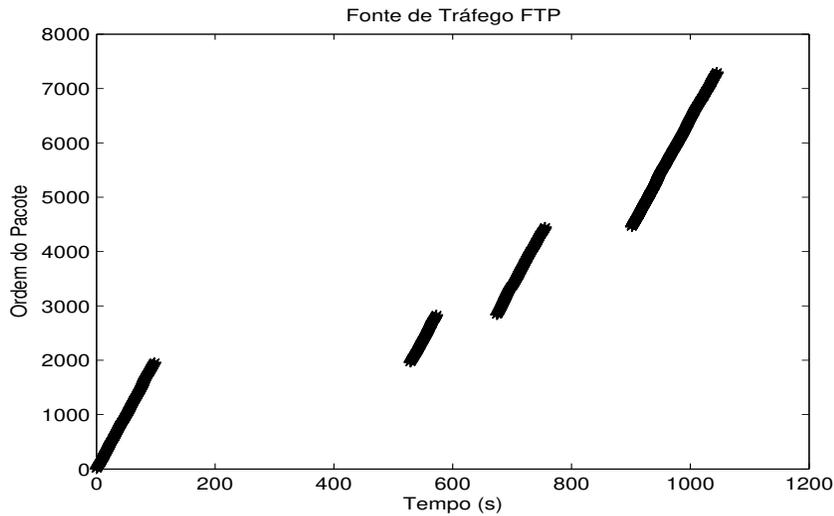


Fig. 3.13: Ordem de Chegadas dos Pacotes do Tráfego FTP

os períodos de inatividade são quase imperceptíveis no gráfico.

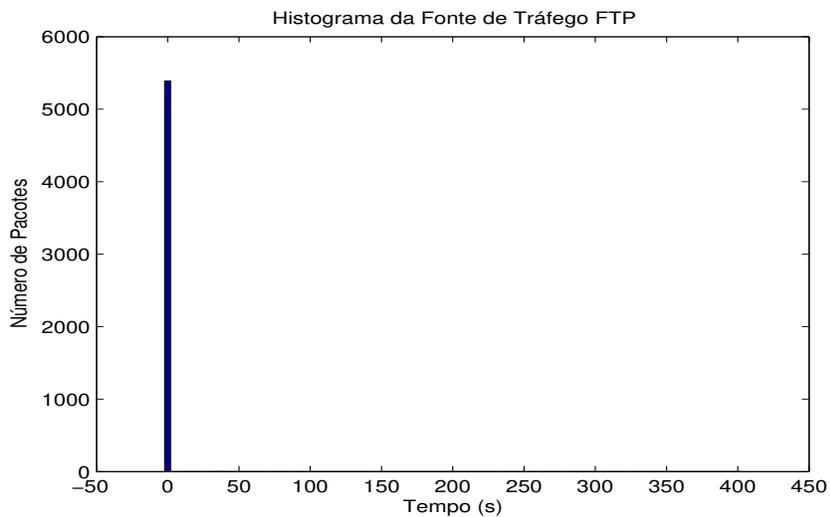


Fig. 3.14: Histograma dos Intervalos entre Chegadas do Tráfego FTP

### 3.5 Modelo WAP

No modelo de tráfego WAP, cada requisição ao navegador é modelada por um pacote de tamanho fixo de 76 bytes. O comprimento do objeto possui distribuição de pareto com média de 256 bytes

e máximo de 1400 bytes sendo os parâmetros desta distribuição  $k$  valendo 71,7 bytes e  $\alpha$  sendo 1,1. Cada requisição gera uma resposta com vários objetos. O número de objetos por resposta é distribuído geometricamente com média de 2 objetos por resposta e o tempo inter-chegadas entre objetos é distribuído exponencialmente com média 1,6 segundos.

O tempo de resposta do *gateway* é o tempo compreendido entre o envio do último byte ao servidor WAP e a resposta do primeiro byte recebido do servidor.

O tempo de leitura é o tempo entre o último objeto recebido e a geração da próxima requisição pelo navegador WAP. A Figura 3.15 ilustra o fluxo de dados do modelo WAP e a Tabela 3.4 descreve os parâmetros adotados neste modelo.

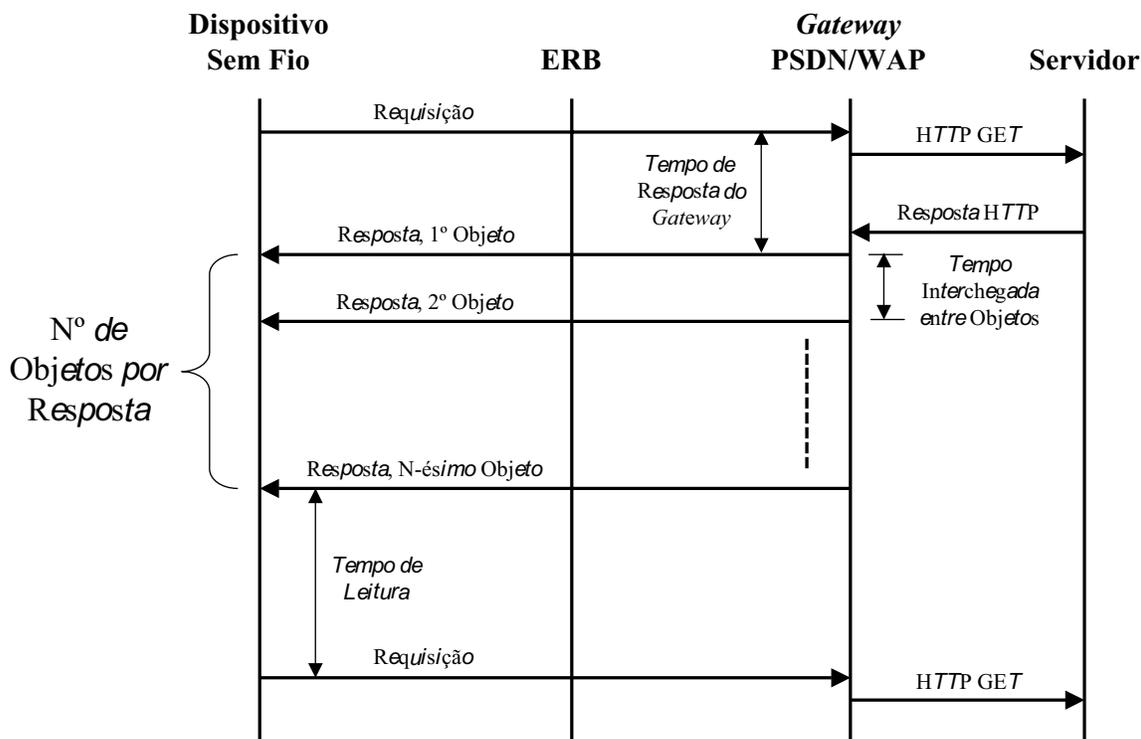


Fig. 3.15: Modelo de Tráfego WAP

### 3.5.1 Histogramas das Fontes WAP

A Figura 3.16 mostra o histograma do comprimento dos pacotes WAP que possuem distribuição de Pareto com média de 256 bytes.

Esta fonte WAP assim como as demais fontes reais HTTP e FTP constituem fontes *On/Off* em que alternam períodos ativos e inativos. Na Figura 3.17 é ilustrada a ordem de chegada dos pacotes das fontes WAP. Nota-se nesta figura os curtos períodos de inatividade quase imperceptíveis.

Tab. 3.4: Parâmetros do Modelo de Tráfego WAP

Parâmetros	Média	Distribuição
Tamanho do Pacote de Requisição WAP	76 bytes	Determinística
Tamanho do Objeto	256 bytes	Pareto
Nº de Objetos por Resposta	2	Geométrica
Tempo Interchegada entre Objetos	1,6 s	Exponencial
Tempo de Resposta do <i>Gateway</i>	2,5 s	Exponencial
Tempo de Leitura	5,5 s	Exponencial

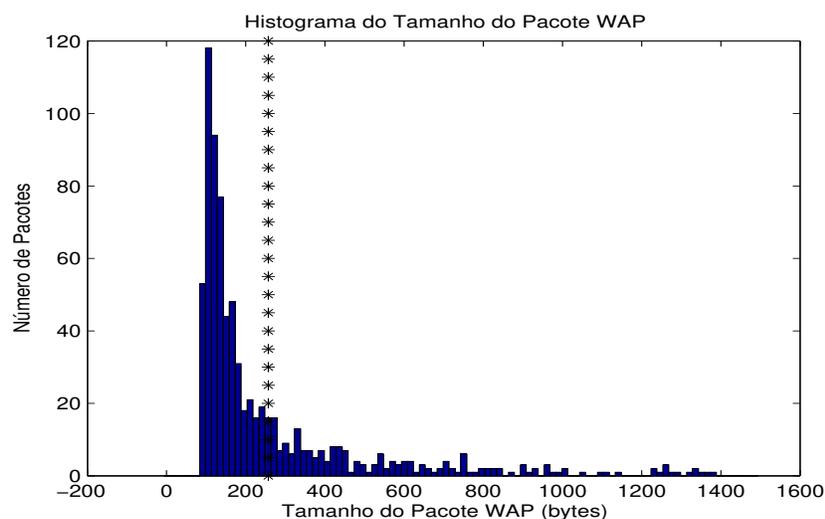


Fig. 3.16: Histograma do Comprimento do Pacote WAP

Na Figura 3.18 é exibido o histograma da distribuição do intervalo (em tempo) entre chegadas de pacotes do tráfego WAP com média aproximada de 3 segundos.

## 3.6 Escalonadores

Nesta seção são apresentados os escalonadores usados neste estudo tendo em vista que a padronização do sistema IS-856 não prescreve os escalonadores com prioridade, WFQ (*Weighted Fair Queuing*) e DRR (*Deficit Round Robin*).

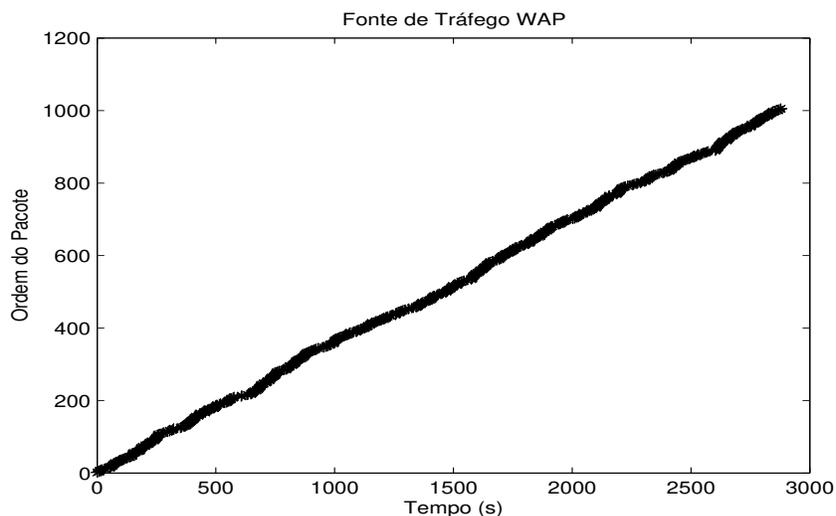


Fig. 3.17: Ordem de Chegada dos Pacotes do Tráfego WAP

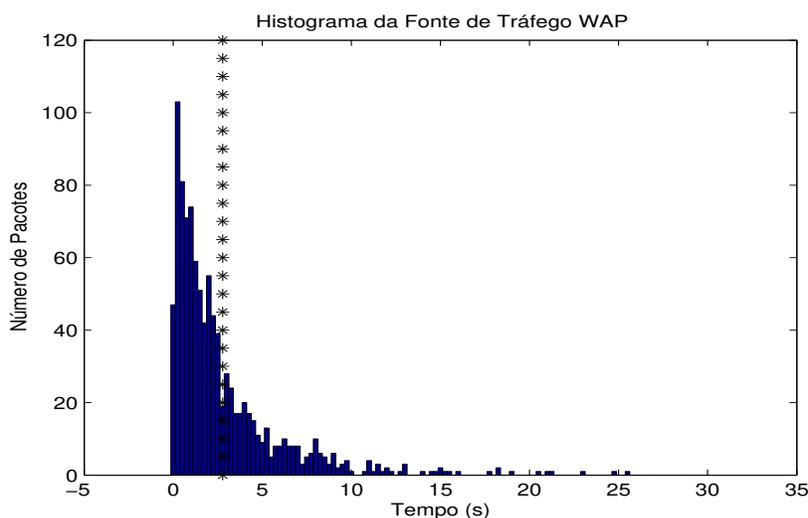


Fig. 3.18: Histograma dos Intervalos entre Chegadas do Tráfego WAP

### 3.6.1 FIFO

O escalonamento FIFO (*First In First Out*) realiza o atendimento dos pacotes que chegam ao servidor de acordo com a ordem de chegada dos pacotes, isto é, o primeiro pacote a chegar é o primeiro a sair do sistema. Neste esquema, os pacotes dos fluxos HTTP, FTP e WAP não são diferenciados entre si, ou seja, nenhum pacote tem prioridade sobre o outro.

### 3.6.2 Prioridade

Já o escalonamento baseado em prioridade atende aos pacotes de acordo com as prioridades atribuídas aos mesmos. Estas prioridades são associadas aos pacotes em algum determinado momento antes de estes chegarem ao servidor do sistema. A atribuição de prioridades aos pacotes é efetuada de acordo ao tipo de fluxo HTTP, FTP e WAP ao qual pertence. Neste esquema os pacotes da fila de menor prioridade somente são atendidos após os pacotes da fila de maior prioridade serem atendidos primeiro. Caso haja somente pacotes com a mesma prioridade em uma fila é transmitido o pacote com maior tempo de espera na fila.

### 3.6.3 WFQ

O outro escalonamento adotado neste estudo, WFQ foi proposto, inicialmente em [28]. Este escalonamento visa o atendimento dos pacotes de forma mais justa possível através da atribuição de uma mesma largura de banda para todos os tipos de pacotes.

Para a realização prática deste escalonamento há diversas implementações disponíveis na literatura, porém para efeito de comparações com os demais escalonamentos abordados neste estudo, foi escolhida a implementação proposta em [29].

Nesta implementação, o escalonamento WFQ é simulado por meio do escalonamento DRR. Desta forma, o servidor do sistema IS-856 trabalha na forma de ciclos em busca de pacotes presentes em cada fila para a sua transmissão.

Os pacotes antes de chegar ao servidor são classificados em filas de acordo com o seu fluxo HTTP, FTP e WAP. Para cada fila são atribuídas cotas com a quantidade de bits a serem transmitidos para aquela fila em questão. Além disso, há também um saldo para cada fila em que a cada ciclo a cota é somada a este saldo.

A transmissão se dá enquanto houver pacotes na fila levando-se em consideração se o saldo é maior que zero e suficiente para a transmissão do pacote. Este saldo é zerado no momento em que não houver mais pacotes na fila em questão. Por fim, para a efetiva implementação do escalonamento WFQ faz-se necessário a atribuição da mesma cota a todas as filas de pacotes presentes no sistema.

#### Pseudo Código para o Algoritmo WFQ

A Figura 3.19 apresenta a principal parte do algoritmo para o escalonamento WFQ.

Neste algoritmo conforme os pacotes chegam ao sistema vão sendo armazenados numa fila de acordo com o seu fluxo. Se há pacotes nesta fila inicia-se um ciclo neste escalonamento somando o valor do saldo a uma certa cota dada em bits.

```

enquanto houver pacotes faça
  se há pacotes na fila então
    ciclo ← ciclo + 1;
    saldo ← saldo + cota;
    enquanto (saldo - tamanho_pacote) > 0
      saldo ← saldo - tamanho_pacote;
      Transmite o pacote;
      i ← i + 1;
      se fila vazia estiver vazia então
        saldo ← 0;
        break;
    fim-se
  fim-enquanto
fim-se
fim-enquanto

```

Fig. 3.19: Pseudo Código WFQ

Na seqüência enquanto houver pacotes na fila e o saldo for suficiente os pacotes são transmitidos. Caso o saldo seja insuficiente para a transmissão do pacote o ciclo corrente é encerrado e inicia novamente outro ciclo somando o saldo remanescente a cota.

Além disso, quando não houver mais pacotes na fila o ciclo atual é terminado, o saldo é zerado e novos pacotes são aguardados na fila para dar início a um novo ciclo.

### Exemplo WFQ

Neste exemplo, o escalonamento WFQ é aplicado a três filas com cotas iguais de 8000 bits para cada classe de fluxo. No primeiro ciclo, todas as variáveis saldo são inicializadas com zero. Na lista de filas ativas um ponteiro indica para qual fila os recursos do sistema estarão disponíveis naquele momento. Neste caso os recursos do sistema estão disponíveis para a fila 1. Como o saldo de 8000 bits é insuficiente para transmitir o pacote de 12000 bits o saldo é acumulado para o próximo ciclo em que os recursos estarão disponíveis para esta fila e encerra-se o ciclo atual. Esta situação é mostrada pela Figura 3.20.

No ciclo 2 a fila ativa passa a ser a fila 2. Nesta fila com o saldo de 8000 bits é transmitido um pacote de 4608 bits. Desta forma, o saldo passa a ser 3392 bits e é acumulado para a transmissão do próximo pacote. Este ciclo é exibido pela Figura 3.21.

Já no ciclo 3 com a fila 3 ativa são transmitidos os pacotes de 1000 bits, 2000 bits, 3000 bits e 2000 bits que no total perfaz uma soma de 8000 bits. Dessa forma, o saldo é zerado e torna-se insuficiente para transmissão do último pacote presente nesta fila tendo como consequência o fim do ciclo corrente ilustrado pela Figura 3.22.

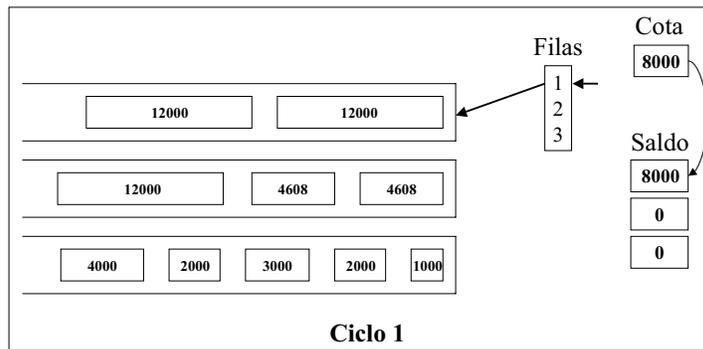


Fig. 3.20: Ciclo 1 WFQ

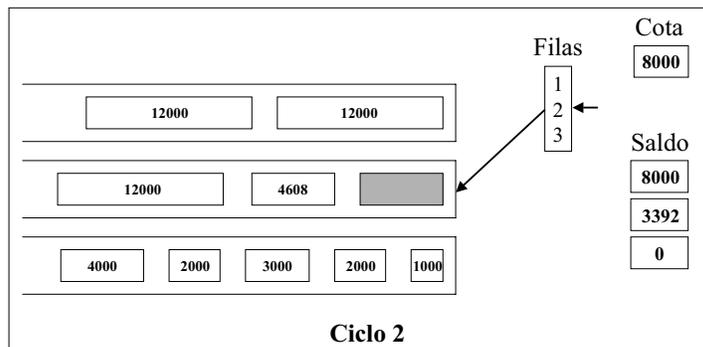


Fig. 3.21: Ciclo 2 WFQ

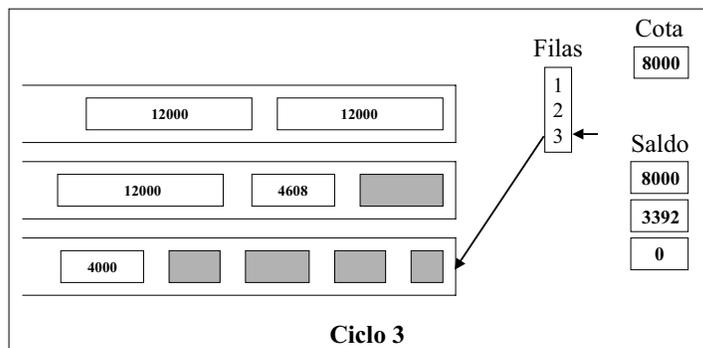


Fig. 3.22: Ciclo 3 WFQ

No ciclo 4 a fila ativa volta a ser fila 1. O saldo remanescente de 8000 bits é somado a cota de 8000 bits. Dessa maneira, o saldo passa a ser 16000 bits viabilizando a transmissão do primeiro pacote de 12000 bits. Após esta transmissão o saldo passa a ser de 4000 bits e como não é possível a transmissão do último pacote de 12000 bits o saldo fica acumulado para o próximo ciclo desta fila. A

Figura 3.23 ilustra esta situação.

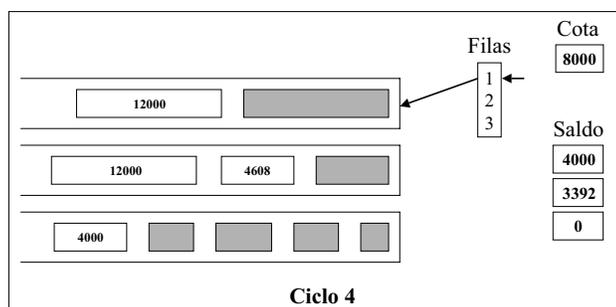


Fig. 3.23: Ciclo 4 WFQ

Na seqüência a fila ativa passa ser a fila 2. O saldo anterior é somado a cota de 8000 bits resultando num saldo de 11392 bits. Por meio deste saldo é transmitido mais um pacote desta fila. O saldo restante é 6784 bits e fica acumulado para o próximo ciclo desta fila uma vez que resta apenas mais um pacote na fila. Este ciclo é mostrado pela Figura 3.24.

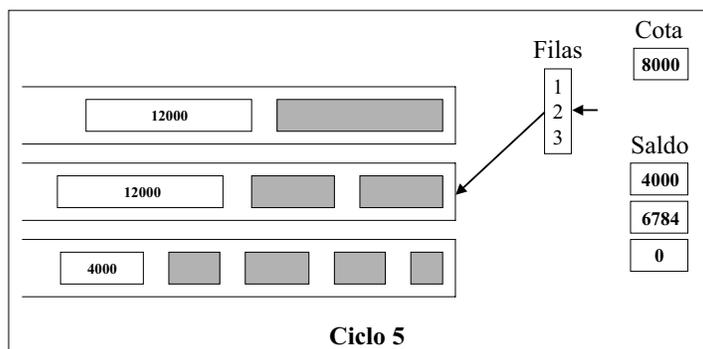


Fig. 3.24: Ciclo 5 WFQ

No ciclo 6 o saldo somado a cota denota um total de 8000 bits. Dessa forma, o pacote de 4000 bits é transmitido e como não há mais pacotes nesta fila o saldo é zerado e a lista de filas ativas passa a contar somente com as filas 1 e 2. A Figura 3.25 exibe este cenário.

No ciclo 7 é transmitido o último pacote da fila 1 tendo em vista que o saldo de 12000 bits garante a transmissão do pacote. Sendo este pacote o último a ser transmitido o saldo é zerado e partir deste ciclo somente há uma fila com pacotes a serem transmitidos. A Figura 3.26 ilustra este ciclo.

Por fim, o último pacote do sistema presente na fila 2 é transmitido. Desta forma, o saldo é zerado e nenhuma fila permanece ativa, pois, não há nenhum pacote no sistema. O sistema fica inoperante até o momento em que chegar algum pacote nas filas de espera podendo neste caso no máximo ter 3 filas ativas ao mesmo tempo. A Figura 3.27 mostra esta situação.

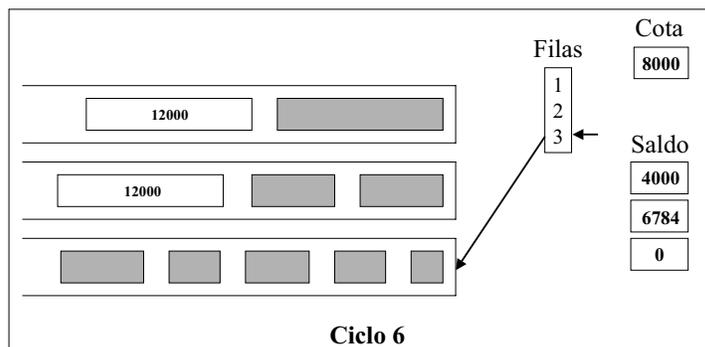


Fig. 3.25: Ciclo 6 WFQ

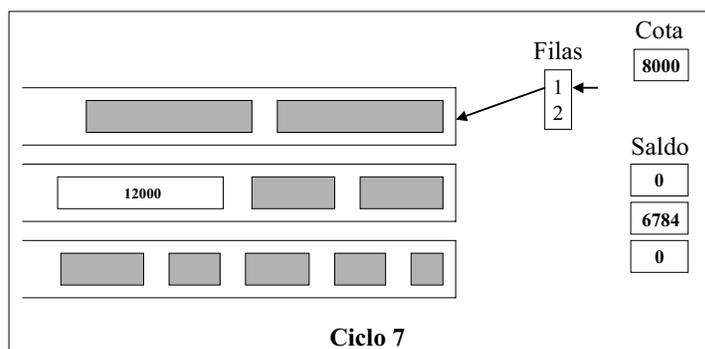


Fig. 3.26: Ciclo 7 WFQ

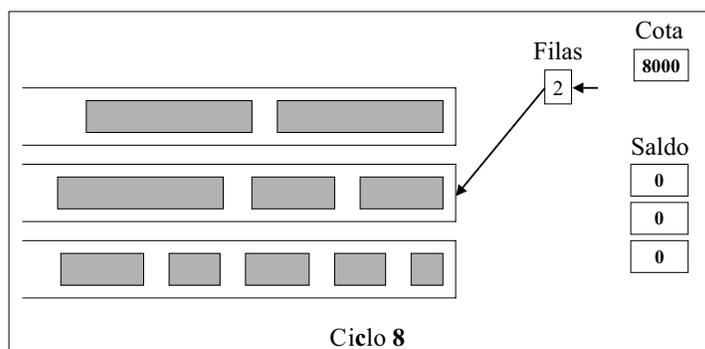


Fig. 3.27: Ciclo 8 WFQ

### 3.6.4 DRR

O escalonamento DRR permite atribuir a cada fila de pacotes largura de banda diferenciada. Em outras palavras este escalonador permite privilegiar determinados tipos de pacotes através de dife-

rentes valores de cotas atribuídos para cada fila (classe de serviço) sendo esta a principal diferença entre os escalonamentos WFQ e DRR nesta implementação específica.

O escalonamento WFQ proposto em [29] é o caso especial do algoritmo DRR em que as cotas são iguais para todos os fluxos. O modelo DRR por sua vez utiliza diferentes cotas para cada tipo de fluxo.

## 3.7 Modelagem do Sistema IS-856

### 3.7.1 Modelo IS-856

A parte em estudo do sistema IS-856 é mostrada na Figura 3.28a. Os pacotes gerados na rede núcleo são enviados até um *buffer* da ERB e ficam esperando até o atendimento. O atendimento (ou a transmissão) dos pacotes é feito de acordo com a Tabela 2.1. Dessa maneira, o número de TSs utilizados para transmitir um pacote varia de acordo com a taxa de dados. Os pacotes são eventualmente segmentados para acomodarem nas taxas que serão enviados. Assim, o modelo do enlace direto do sistema IS-856 pode ser representado conforme a Figura 3.28b.

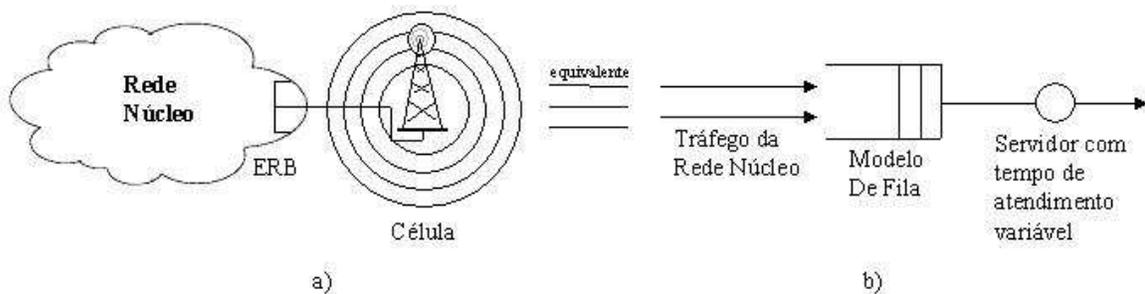


Fig. 3.28: Modelo IS-856

Com o objetivo de avaliar o impacto de diversos tipos de escalonamentos como FIFO, com prioridade, WFQ e DRR no sistema IS-856 foram utilizadas várias filas. Para cada fila somente um tipo de tráfego entre os propostos neste estudo é permitido. O número de filas varia de acordo com cenários propostos.

### Medidas de Desempenho

As medidas de desempenho consideradas neste estudo são: atraso médio dos pacotes, vazão de saída, percentual de perda de cada tipo de tráfego e utilização do sistema. Para cada fila são avaliados

o atraso, a vazão de saída e em alguns cenários o percentual de perda. Já o fator de utilização considera o número de filas existentes em cada cenário.

### Classes de Cobertura

As classes de coberturas adotadas foram as mesmas usadas em [4] em que foi feita uma estimativa aproximada (não há dados disponíveis na literatura) da distribuição de probabilidade destas classes e são mostradas na Tabela 3.5.

Tab. 3.5: Distribuição de Probabilidade pela Taxa de Dados

<b>Taxa (kbps)</b>	<b>Probabilidade</b>
<b>38,4</b>	2%
<b>76,8</b>	3%
<b>153,6</b>	8%
<b>307,2</b>	18%
<b>614,4</b>	23%
<b>921,6</b>	17%
<b>1228,8</b>	15%
<b>1843,2</b>	10%
<b>2457,6</b>	4%

### Suposições do Modelo

As seguintes suposições são adotadas para a modelagem do sistema IS-856:

1. Em determinado momento os pacotes são classificados de acordo com o tipo de tráfego HTTP, WAP e FTP e associados a uma fila antes de chegar ao escalonador.
2. No caso do escalonamento com prioridade além desta classificação são associadas prioridades aos pacotes também, antes de chegar ao escalonador.
3. O escalonador utiliza algum dos seguintes esquemas: FIFO, com prioridade, WFQ e DRR.
4. Para o caso do escalonador FIFO os pacotes são atendidos por ordem de chegada.
5. Para o escalonamento com prioridade os pacotes da fila de menor prioridade somente são atendidos após o atendimento de todos os pacotes da fila de maior prioridade. Além disso, o escalonamento com prioridade é considerado sem preempção, ou seja, o atendimento de um pacote

de menor prioridade prossegue até o fim, independentemente da chegada de uma pacote com maior prioridade na fila de espera.

6. Para os outros casos WFQ e DRR a transmissão de pacotes é limitada pela largura de banda destinada a cada fila.
7. O *buffer* de cada fila pode ter tamanho infinito e finito variando de 10 a 100 pacotes armazenados no *buffer*. Os pacotes são armazenados no esquema FIFO.
8. Somente o tráfego do enlace direto é avaliado.
9. Para a simulação foi avaliado o comportamento de 20000 pacotes para fontes poissonianas e 40000 pacotes para as fontes reais HTTP, FTP e WAP. O intervalo de pacotes é maior das fontes reais em razão destas possuírem um período transitório maior.
10. Para a modelagem de tráfegos adotaram-se os modelos poissonianos e os modelos reais descritos em [20].

### 3.7.2 Cenários de Análise

Nas simulações foram variadas as concentrações dos tráfegos HTTP, FTP e WAP de forma a avaliar o impacto destas fontes no sistema IS-856. Além disso, aliada a estas fontes foram empregados diferentes esquemas de escalonamento neste mesmo sistema a fim de averiguar possíveis melhoras no desempenho. Assim, foram simulados cinco cenários com diferentes concentrações de tráfegos e diversos tipos de escalonamentos.

#### Cenário 1

No primeiro cenário exibido na Tabela 3.6, as fontes WAP e HTTP são distribuídas 70% e 30%, respectivamente, sendo a fonte HTTP prioritária. Neste cenário a distribuição dos tráfegos indica o início da utilização dos serviços 3G como o HTTP, porém como está no início os serviços 2,5G representam a maioria do tráfego total. Além do uso das fontes reais citadas acima, também foram modeladas fontes poissonianas equivalentes às fontes reais. A taxa média de chegada de pacotes das fontes poissonianas assim como o tamanho médio dos pacotes destas fontes são distribuídos exponencialmente. Para a simulação destas fontes poissonianas foram consideradas a taxa média de chegada de pacotes (pacotes/segundo) das fontes reais e também o tamanho médio dos pacotes (bits) das fontes reais. Desta forma, as fontes poissonianas são avaliadas neste cenário e nos próximos três cenários a seguir.

Tab. 3.6: Cenário 1

<b>Tipo de Tráfego</b>	<b>Nível de Prioridade</b>	<b>Probabilidade</b>
<b>HTTP</b>	1	30%
<b>WAP</b>	2	70%

### Cenário 2

Para o segundo cenário ilustrado na Tabela 3.7, as fontes são distribuídas em 30% WAP e 70% HTTP mantendo a maior prioridade da fonte HTTP. Já neste cenário a tecnologia 3G estaria amadurecida e a maior parte do tráfego total seria de serviços 3G como o HTTP. Neste cenário também é avaliado o comportamento do sistema IS-856 face a proporção majoritária dos pacotes prioritários HTTP.

Tab. 3.7: Cenário 2

<b>Tipo de Tráfego</b>	<b>Nível de Prioridade</b>	<b>Probabilidade</b>
<b>HTTP</b>	1	70%
<b>WAP</b>	2	30%

### Cenário 3

Já no terceiro cenário mostrado na Tabela 3.8 é acrescentada a fonte FTP às fontes HTTP e WAP. Em razão da fonte FTP ter um grande peso no volume de tráfego, as fontes ficaram distribuídas em 40% WAP, 10% HTTP e 50% FTP. A fonte HTTP tem a maior prioridade, a fonte FTP prioridade intermediária e WAP a menor prioridade. Este cenário representa uma transição dos serviços 2,5G para 3G, pois, o tráfego WAP constitui 40% do tráfego total.

Tab. 3.8: Cenário 3

<b>Tipo de Tráfego</b>	<b>Nível de Prioridade</b>	<b>Probabilidade</b>
<b>HTTP</b>	1	10%
<b>WAP</b>	2	40%
<b>FTP</b>	3	50%

### Cenário 4

No quarto cenário exibido na Tabela 3.9, as fontes estão distribuídas em 10% WAP, 40% HTTP e 50% FTP possuindo maior prioridade a fonte HTTP, a fonte FTP prioridade intermediária e WAP a menor prioridade. Neste cenário efetivamente os serviços 3G representam 90% do tráfego total.

Tab. 3.9: Cenário 4

<b>Tipo de Tráfego</b>	<b>Nível de Prioridade</b>	<b>Probabilidade</b>
<b>HTTP</b>	1	40%
<b>WAP</b>	2	10%
<b>FTP</b>	3	50%

### Cenário 5

Por fim, para o último cenário exibido na Tabela 3.10 as fontes estão distribuídas em 60% HTTP, 10% WAP e 30% FTP. Para o escalonamento com prioridade a fonte HTTP tem a prioridade mais alta, a fonte WAP prioridade intermediária e a fonte FTP a menor prioridade. Para os outros escalonamentos WFQ as fontes HTTP, WAP e FTP possuem cota de 8000 bits e para o escalonamento DRR a fonte HTTP tem cota de 16000 bits, a fonte WAP cota de 8000 bits e a fonte FTP cota de 4000 bits. Assim, são comparadas as vantagens e desvantagens dos escalonamentos com prioridade, WFQ e DRR.

Tab. 3.10: Cenário 5

<b>Tráfego</b>	<b>Probabilidade</b>	<b>Prioridade</b>	<b>Cota WFQ</b>	<b>Cota DRR</b>
HTTP	60%	1	8000	16000
WAP	10%	2	8000	8000
FTP	30%	3	8000	4000

## 3.8 Conclusão

Os modelos reais de tráfego HTTP, FTP e WAP descritos neste capítulo representam de forma bastante satisfatória os tráfegos no mundo real uma vez que os modelos propostos em [20] foram obtidos face a amplos estudos presente na literatura e medidas de desempenho do mundo real. Desta

forma, os modelos reais de tráfego constituem um bom parâmetro para a avaliação de desempenho do sistema IS-856.

Além destas fontes reais, as fontes poissonianas também foram avaliadas em alguns cenários. Em virtude das fontes poissonianas serem fontes amplamente conhecidas não foram detalhadas neste capítulo as mesmas.

Finalizando este capítulo são descritos os parâmetros necessários para as simulações, as suposições adotadas e os cenários de simulações. Além disso, os cenários propostos neste estudo visam avaliar certas circunstâncias das redes de telefonia celular em que o uso dos escalonamentos com prioridade, WFQ e DRR em um ambiente de tráfegos reais permitam assegurar QoS a seus usuários.

# Capítulo 4

## Análise dos Resultados

### 4.1 Introdução

Para os quatro primeiros cenários introduzidos na seção 3.7.2, os resultados obtidos nas simulações estão apresentados nos gráficos a seguir. Nestes gráficos a sigla SPr refere-se ao escalonamento sem prioridade, a sigla Ps identifica os tráfegos poissonianos e as siglas PrX associam prioridades a cada tipo de tráfego variando de Pr1 a Pr3, isto é, da prioridade mais alta para a mais baixa. Os tráfegos poissonianos são gerados levando-se em conta as mesmas taxas (pacotes/segundo) dos tráfegos reais. Além disso, as legendas destes gráficos estão dispostas na mesma ordem das curvas de dados para facilitar a leitura.

Para o cenário 5, os resultados também são exibidos nos gráficos a seguir. Assim como nos cenários anteriores as legendas destes gráficos são exibidas na mesma ordem das curvas de dados para facilitar a leitura. Nestes gráficos a sigla BI refere-se a *buffer* infinito, e a sigla BF X significa *buffer* finito e X representa o tamanho máximo deste *buffer* limitado. A sigla FIFO identifica o escalonamento FIFO. As siglas PrX associam prioridades a cada tipo de tráfego variando de Pr1 a Pr3, isto é, do tráfego mais prioritário para o menos prioritário. Por fim, as siglas WFQX e DRRX referem-se aos escalonamentos WFQ e DRR em que X representa a cota para cada um destes escalonamentos.

### 4.2 Cenário 1

Para a simulação do sistema IS-856 sem prioridade (SPr), considerando 70% do tráfego WAP e os outros 30% HTTP (cenário 1) a vazão variou de 105 kbps a 412 kbps em função do carregamento do sistema. Para o mesmo cenário, porém, considerando o seu equivalente poissoniano sem prioridade (SPr Ps) a vazão apresentou uma ligeira alta de 4% em relação à situação anterior. Além disso, a vazão do escalonamento sem prioridade mostrou-se igual à soma das vazões do escalonamento com

prioridade. Na média, o desvio padrão foi 3,69% do valor da média enquanto o intervalo de confiança de 95% apresentou 2,29% do valor da média.

Considerando-se escalonamento com prioridade pode ser observada a maior vazão do tráfego WAP Pr2 em relação à fonte HTTP Pr1 prioritária mesmo sendo de menor prioridade, pois, na maioria das vezes o enlace direto encontra-se livre face ao baixo volume de tráfego apresentado pela fonte HTTP Pr1. Já com relação aos equivalentes poissonianos a fonte WAP Ps Pr2 apresenta vazão igual ou superior a sua correspondente real WAP Pr2, enquanto a fonte HTTP Ps Pr1 poissoniana apresenta uma vazão menor em comparação à da fonte real HTTP Pr1 devido à baixa concentração de tráfego. Na média, o desvio padrão foi 6,84% do valor da média enquanto o intervalo de confiança de 95% apresentou 4,24% do valor da média.

A Figura 4.1 exibe a vazão versus a ocupação do sistema IS-856 para o cenário 1.

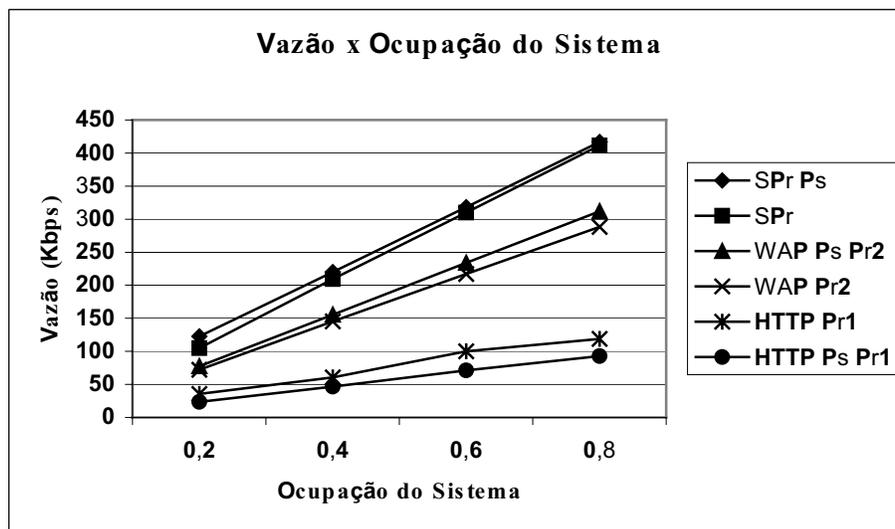


Fig. 4.1: Vazão em Função da Ocupação do Sistema para o Cenário 1

Com relação ao atraso médio dos pacotes neste cenário 1, nota-se que para 20% do carregamento do sistema o atraso médio dos pacotes com ou sem prioridade é em torno de 10 ms em virtude do baixo volume de tráfego presente neste ponto inicial e da alta taxa de vazão de entrada do tráfego WAP. Já com o aumento do fator de utilização do sistema variando de 20% até 80% torna-se evidente os benefícios da priorização de tráfego em virtude do baixo atraso mostrado pelo tráfego prioritário HTTP Pr1. O tráfego WAP Pr2 menos prioritário e o escalonamento sem prioridade (SPr) apresentaram atrasos superiores a 1s.

Para todos os tráfegos poissonianos os atrasos médios dos pacotes foram inferiores a 100 ms. As fontes poissonianas são fontes descorrelacionadas enquanto as fontes reais WAP, HTTP são correlacionadas apresentando, dessa maneira, altos atrasos.

A Figura 4.2 mostra o atraso médio dos pacotes em função da utilização do sistema.

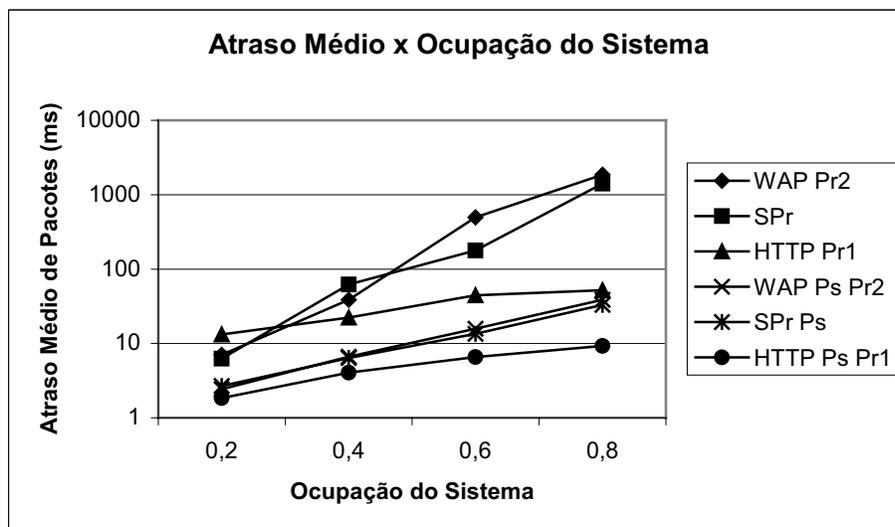


Fig. 4.2: Atraso Médio em Função da Ocupação do Sistema para o Cenário 1

### 4.3 Cenário 2

As distribuições do tráfego no cenário 2 foram alteradas para 70% HTTP e 30% WAP e o desempenho da vazão em função da ocupação do sistema é mostrada na Figura 4.3. Como pode ser visto a fonte HTTP Pr1 prioritária em virtude do seu maior volume de tráfego possui maior vazão do que WAP Pr2 em toda a variação de ocupação do sistema. Com relação ao escalonamento sem prioridade SPr a vazão do sistema permaneceu praticamente a mesma do cenário anterior variando de 101 kbps até 431 kbps, ou seja, para diferentes concentrações de tráfego este tipo de escalonamento obteve o mesmo desempenho. Na média, o desvio padrão foi 4,78% do valor da média enquanto o intervalo de confiança de 95% apresentou 2,96% do valor da média.

O atraso médio dos pacotes para o cenário 2 em função da ocupação do sistema é exibido na Figura 4.4. Nesta figura observa-se para carregamento de 20% do sistema todas as fontes exceto as poissonianas possuem o mesmo atraso utilizando ou não o escalonamento com prioridade. Porém, a medida que aumenta a ocupação do sistema chegando-se 80% verifica-se o melhor desempenho do escalonamento com prioridade que privilegia tanto tráfego real HTTP Pr1 quanto o poissoniano HTTP Ps Pr1. Nota-se também o aumento do atraso médio em relação ao cenário 1 devido a maior concentração do tráfego HTTP sendo este mais "pesado" que o tráfego WAP.

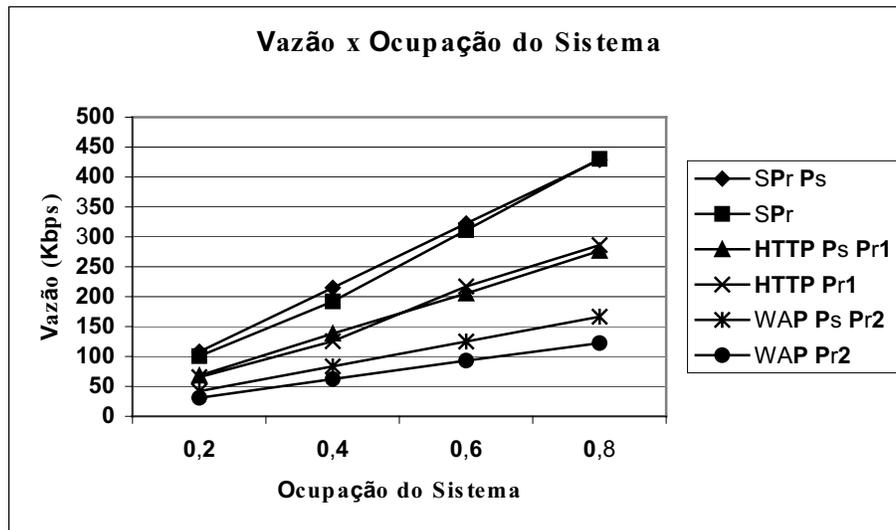


Fig. 4.3: Vazão em Função da Ocupação do Sistema para o Cenário 2

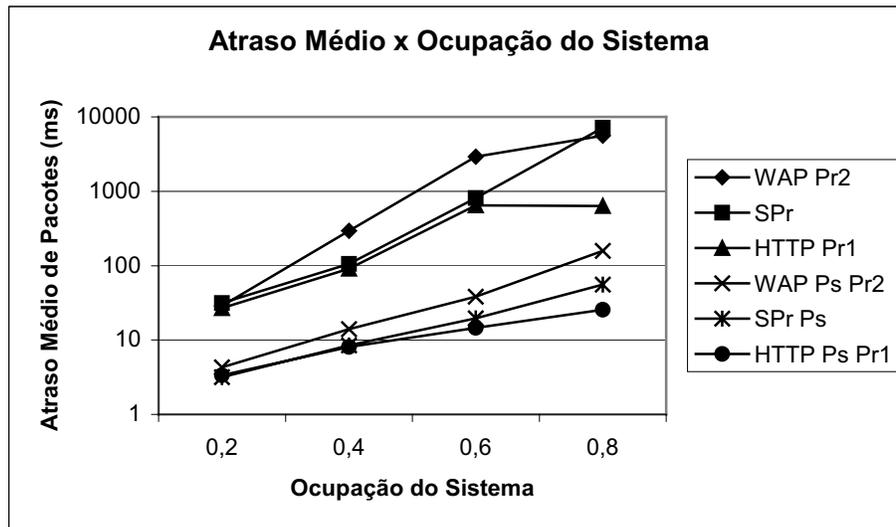


Fig. 4.4: Atraso Médio em Função da Ocupação do Sistema para o Cenário 2

## 4.4 Cenário 3

Para o cenário 3 foi acrescentada a fonte FTP que se caracteriza por altas taxas de vazões e longos atrasos. Como pode ser observada na Figura 4.5, a fonte FTP possui uma alta taxa de vazão para o carregamento do sistema variando de 30% a 90%. Além disso, a fonte WAP também possui este comportamento em razão dos 40% do volume de tráfego e da sua alta taxa de vazão de entrada. A vazão da fonte HTTP apresenta uma pequena taxa pelo fato do baixo volume de tráfego de 10%.

Além disso, neste cenário as vazões dos escalonamentos sem prioridade SPr poissoniano ou real aproximam-se dos 500 kbps devido a maior concentração de tráfegos do modelo TCP como HTTP e FTP. Na média, o desvio padrão foi 6,68% do valor da média enquanto o intervalo de confiança de 95% apresentou 3,18% do valor da média.

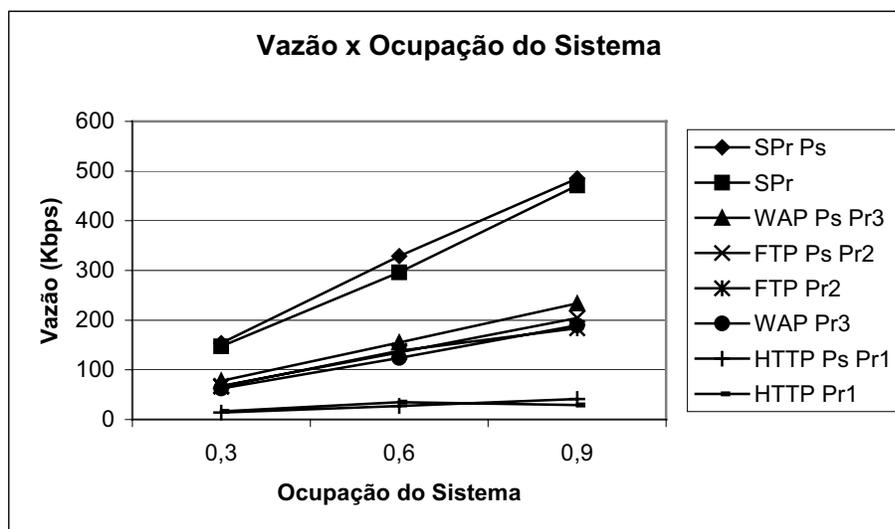


Fig. 4.5: Vazão em Função da Ocupação do Sistema para o Cenário 3

Com referência ao atraso médio dos pacotes a priorização de tráfego mostrou-se neste cenário bem melhor do que o tráfego sem prioridade ilustrada pela Figura 4.6. Variando-se o fator de utilização de 30% a 90% é observada a acentuação das diferenças entre os atrasos médios das fontes mais prioritárias e os atrasos das fontes menos prioritárias. Além disso, a fonte WAP Pr3 com prioridade mais baixa apresenta pior desempenho em razão da sua prioridade e pela alta concentração de tráfegos HTTP Pr1 e FTP Pr2.

## 4.5 Cenário 4

Este cenário apresenta a seguinte distribuição entre as fontes: 10% WAP, 40% HTTP e 50% FTP. A vazão deste cenário é exibida na Figura 4.7. Note que as fontes FTP e HTTP possuem altas taxas de vazão devido as suas próprias características de altas taxas de vazões e a proporção destas fontes do volume total de tráfego. Na média, o desvio padrão foi 4,81% do valor da média enquanto o intervalo de confiança de 95% apresentou 2,98% do valor da média.

A Figura 4.8 mostra o atraso médio dos pacotes em função da ocupação do sistema para o cenário 4. Nesta figura fica evidente o menor atraso dos pacotes HTTP Pr1 com maior prioridade em compa-

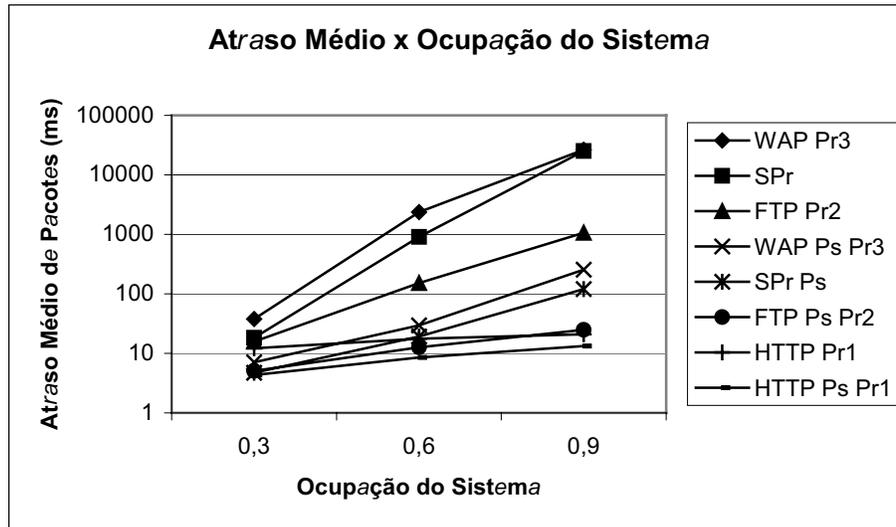


Fig. 4.6: Atraso Médio em Função da Ocupação do Sistema para o Cenário 3

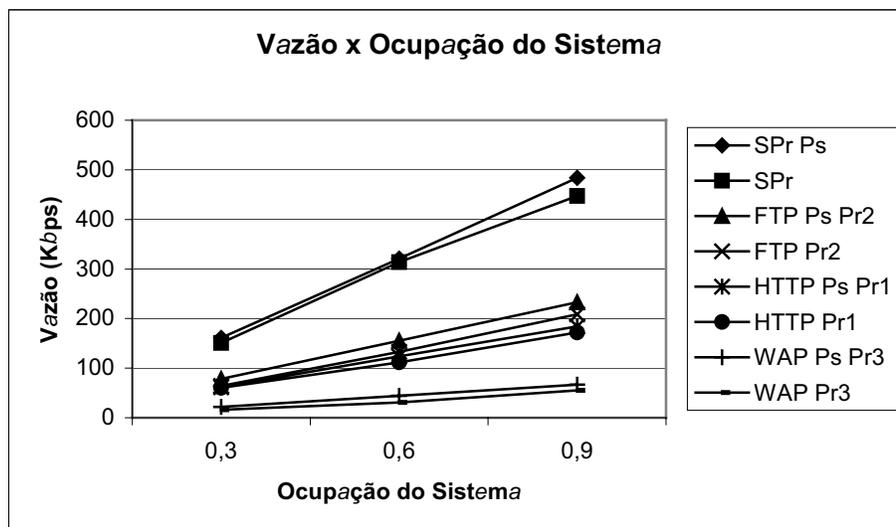


Fig. 4.7: Vazão em Função da Ocupação do Sistema para o Cenário 4

ração as outras fontes menos prioritárias. Porém, em razão da fonte FTP Pr2 possuir longos atrasos a fonte WAP Pr3 de menor prioridade foi prejudicada tendo como consequência atrasos bastante insatisfatórios na variação de 30% a 90% dos pontos do carregamento do sistema.

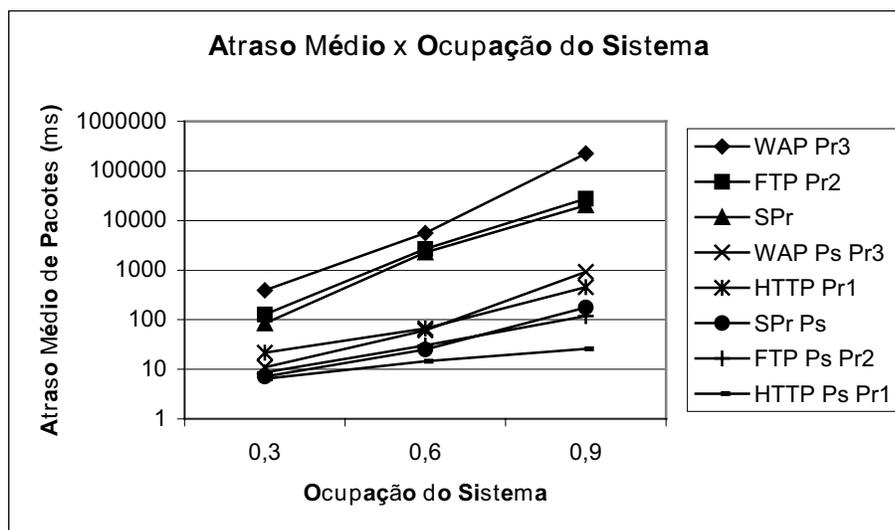


Fig. 4.8: Atraso Médio em Função da Ocupação do Sistema para o Cenário 4

## 4.6 Cenário 5

Por fim, no cenário 5 as fontes estão distribuídas em 60% HTTP, 10% WAP e 30% FTP. Para uma melhor avaliação do impacto dos diferentes escalonamentos propostos os resultados foram divididos por fontes. De uma forma geral as fontes com *buffers* infinitos apresentam as maiores vazões e as fontes com *buffers* finitos vazões menores em razão do descarte de pacotes. Na média neste último cenário, o desvio padrão foi 7,71% do valor da média enquanto o intervalo de confiança de 95% apresentou 5,04% do valor da média.

Na Figura 4.9 a vazão do sistema é apresentada considerando-se somente a fonte HTTP. Esta figura mostra que ao longo da variação da ocupação do enlace as vazões com *buffer* infinito tendem a apresentar vazões em torno de 320 kbps enquanto com *buffer* finito de 10 elementos restringe as vazões a 200 kbps. As exceções a esta regra são as vazões do escalonamentos com prioridade Pr1 com *buffer* infinito e DRR com *buffer* finito mostrado na figura pela sigla BF10 DRR16. A vazão do escalonamento com prioridade e *buffer* infinito apresenta vazão de aproximadamente 250 kbps em razão do menor volume de tráfego apresentados nas simulações deste cenário em relação aos outros escalonamentos. Já a vazão do escalonamento DRR16 com *buffer* limitado apresenta uma vazão acima da média dos outros escalamentos em virtude da maior reserva de banda alocada com a cota de 16000 bits.

Na Figura 4.10 é mostrado o comportamento do sistema com relação a fonte WAP e as vazões dos diferentes escalonamentos avaliados. As vazões desta figura basicamente apresentam o mesmo comportamento das vazões das fontes HTTP.

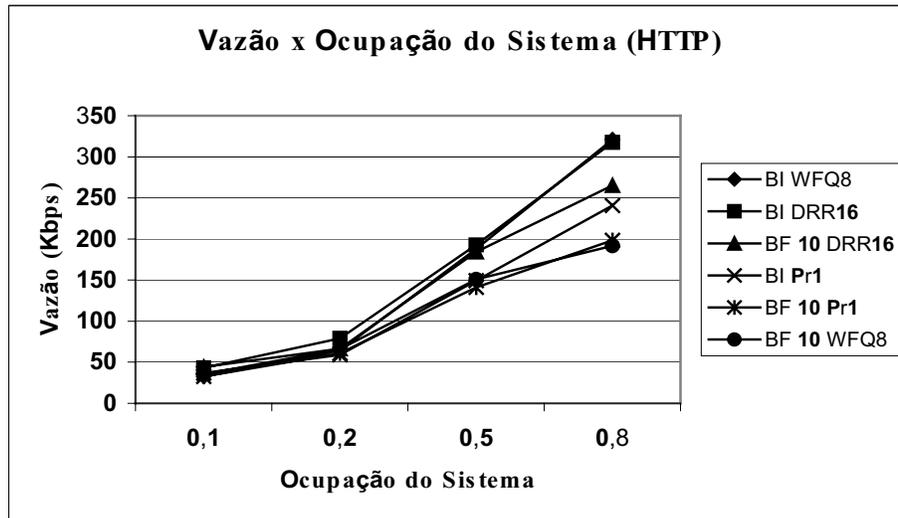


Fig. 4.9: Vazão HTTP em função da Ocupação do Sistema para o Cenário 5

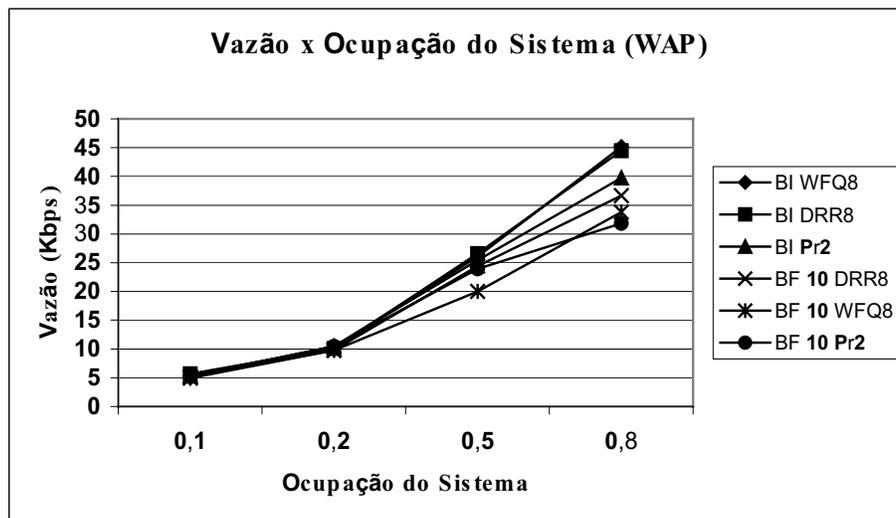


Fig. 4.10: Vazão WAP em função da Ocupação do Sistema para o Cenário 5

As vazões do tráfego FTP são exibidas na Figura 4.11. Nesta figura os percentuais de ocupação de enlace iniciais foram omitidos em razão das fontes FTP alocarem mais banda do que as outras fontes. Nesta figura as vazões dos escalonamentos com *buffer* infinito concentraram-se em torno de 150 kbps enquanto as vazões com *buffer* finito situaram-se em 95 kbps.

Nas figuras seguintes, para uma melhor análise do comportamento do sistema IS-856, os resultados dos atrasos médios e percentuais de perda de pacotes são mostrados para cada fonte HTTP, WAP e FTP separadamente.

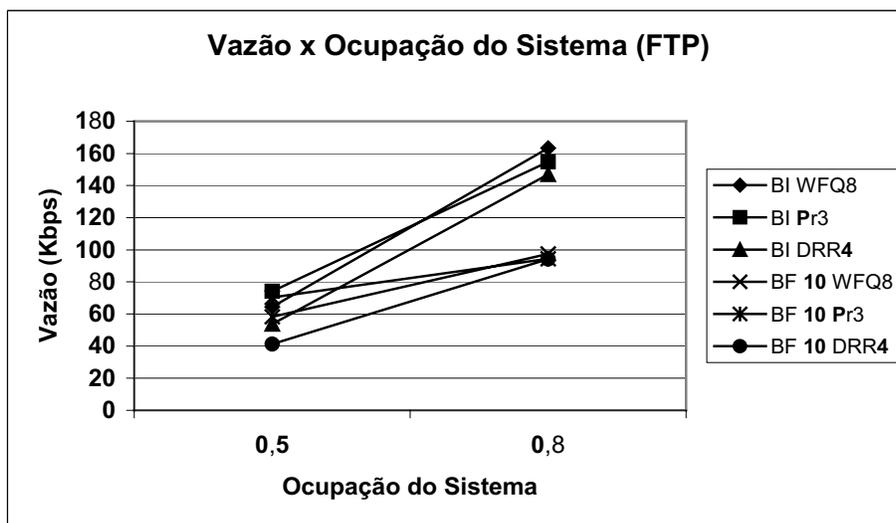


Fig. 4.11: Vazão FTP em função da Ocupação do Sistema para o Cenário 5

A Figura 4.12 mostra o atraso médio do sistema IS-856 em função do carregamento do sistema variando de 10% a 80% somente para a fonte HTTP. Pode ser observado nesta figura que para o escalonamento com a prioridade mais alta Pr1 e o escalonamento DRR com cota de 16000 bits são obtidos atrasos muito próximos. Entretanto, para o escalonamento WFQ com cota de 8000 bits, obteve-se piores atrasos chegando a até 10 s. Além disso, ao limitar o *buffer* as diferenças entre os atrasos diminuem consideravelmente porém, os escalonamentos com prioridade e DRR16 levaram vantagem sobre o escalonamento WFQ8.

O percentual de perda de pacotes HTTP em função da ocupação do sistema IS-856 é mostrado na Figura 4.13. Nesta figura o escalonamento WFQ8 apresenta mais pacotes descartados em relação aos escalonamentos com prioridade Pr1 e DRR16, acentuando-se conforme aumenta ocupação do sistema. Enquanto os escalonamentos com prioridade Pr1 e DRR16 apresentaram descartes semelhantes.

Para este mesmo cenário 5 mudando o foco para o atraso médio dos pacotes WAP em função da ocupação do sistema IS-856 é exibida a Figura 4.14. Como pode ser observado o escalonamento com prioridade intermediária Pr2 apresenta atrasos muito superiores aos outros escalonamentos WFQ8 e DRR8, acentuando-se com o aumento da ocupação do sistema. Os escalonamentos WFQ8 e DRR8 apresentam diferenças mínimas entre uso de *buffers* limitados ou não devido aos 10% do tráfego. Além disso, os comportamentos dos escalonamentos WFQ e DRR são praticamente os mesmos em razão da utilização da mesma largura de banda.

Com referência ao percentual de descartes de pacotes WAP o escalonamento com prioridade intermediária mostra-se muito pior do que os outros escalonadores WFQ8 e DRR8 uma vez que estes

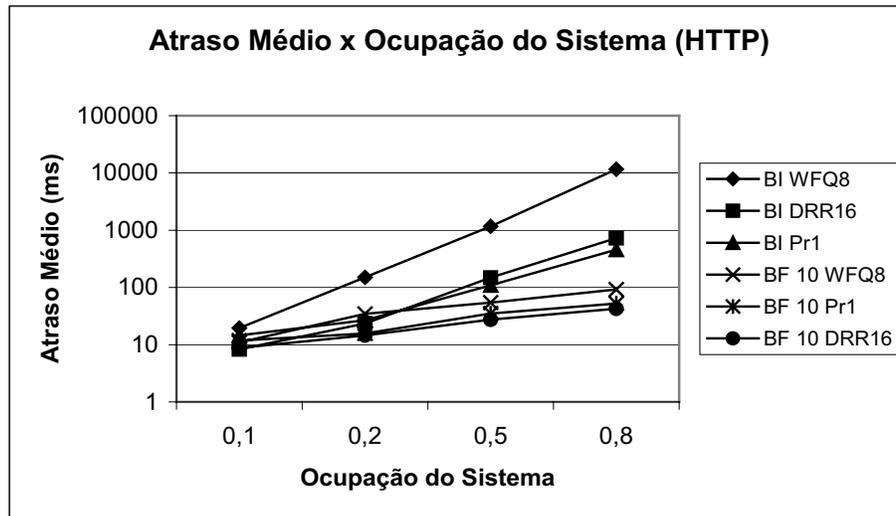


Fig. 4.12: Atraso Médio das fontes HTTP em função da Ocupação do Sistema para o Cenário 5

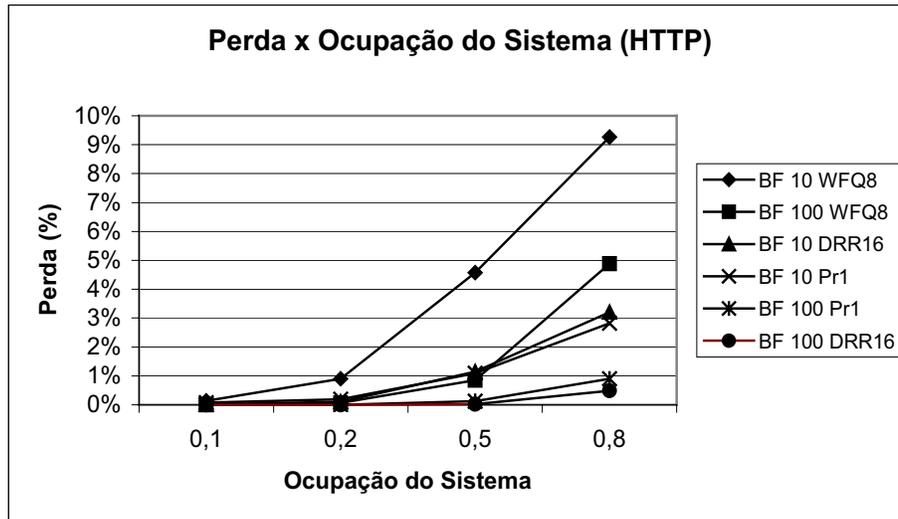


Fig. 4.13: Percentual de perda de pacotes HTTP em função da Ocupação do Sistema para o Cenário 5

escalamentos não apresentam descarte de pacotes em virtude da baixa concentração de tráfego WAP.

Infelizmente devido a uma única fonte FTP ocupar uma banda mais larga do que as outras fontes, não foi possível a obtenção de alguns pontos desta figura. Mas, com base nos pontos obtidos na simulação, a Figura 4.16 exibe o atraso médio dos pacotes FTP versus a ocupação do sistema. Como pode ser observado o escalonamento com baixa prioridade Pr3 com *buffer* infinito obtém o pior atraso

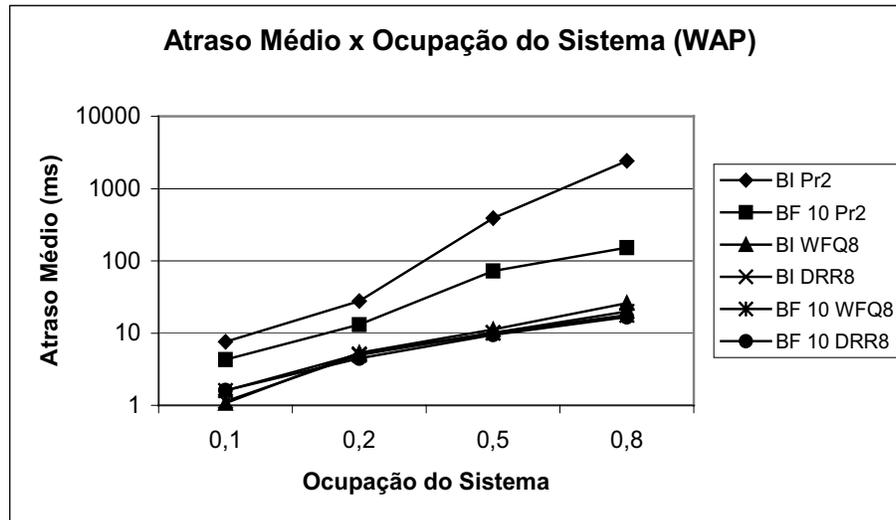


Fig. 4.14: Atraso Médio das fontes WAP em função da Ocupação do Sistema para o Cenário 5

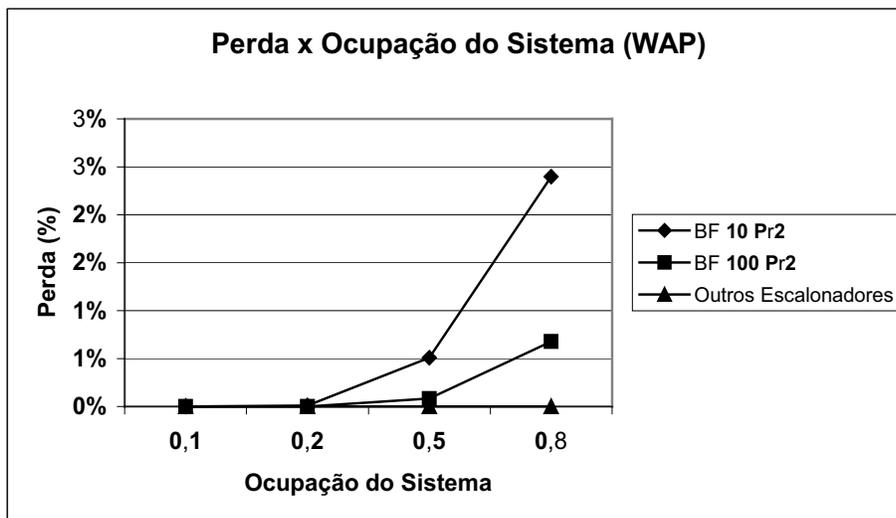


Fig. 4.15: Percentual de perda de pacotes WAP em função da Ocupação do Sistema para o Cenário 5

entre os escalonamentos DRR4 e WFQ8. Considerando o *buffer* limitado de 10 pacotes, a diferença entre os escalonamentos foram atenuadas, porém o WFQ8 obtém o menor atraso devido a reserva de uma maior largura de banda.

Por fim, a Figura 4.17 exibe o percentual de perda de pacotes FTP em função da ocupação do sistema. Nesta figura torna-se evidente a alta taxa de perda de pacotes FTP com o escalonamento Pr3 chegando a atingir até 8% de perda de pacotes. Enquanto os escalonamentos que reservam banda como WFQ8 e DRR4 apresentam pequenas taxas de perdas de pacotes, com escalonamento WFQ8

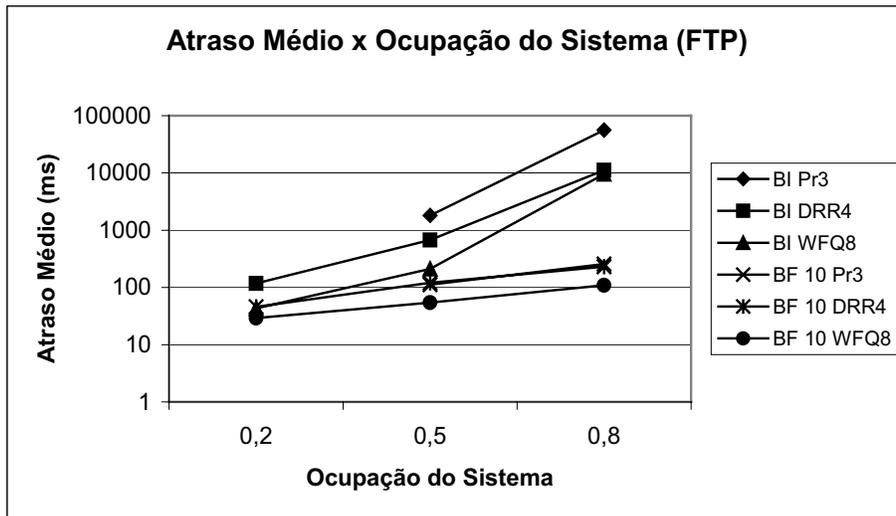


Fig. 4.16: Atraso Médio das fontes FTP em função da Ocupação do Sistema para o Cenário 5

apresentando perda bem menores que o escalonamento DRR4.

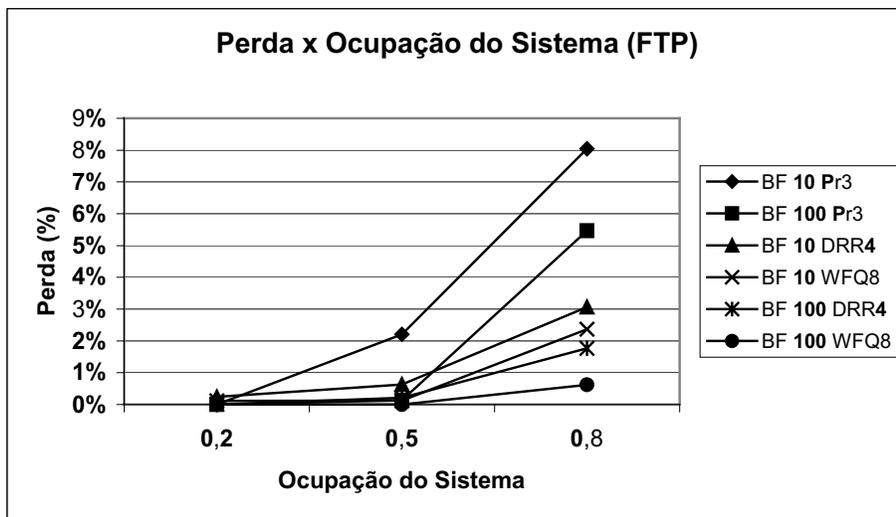


Fig. 4.17: Percentual de perda de pacotes FTP em função da Ocupação do Sistema para o Cenário 5

## 4.7 Modelo Teórico Para o Tráfego Poissoniano Sem Prioridade

O modelo teórico adotado nesta seção restringiu-se a apenas ao tráfego poissoniano sem prioridade (FIFO), pois, as fontes reais HTTP, FTP e WAP em virtude das suas elevadas complexidades

inviabilizam a obtenção de modelos analíticos adequados a estas fontes. Além disso, para o escalonamento com prioridade das fontes poissonianas não foi possível a adoção de um modelo teórico em razão da ocorrência de segmentação dos pacotes quando estes chegam ao servidor.

No sistema IS-856 o pacote que chega de acordo com o processo poissoniano é segmentado em um ou vários pacotes menores e são transmitidos através de um ou mais *slots* ou canais. Por exemplo: quando chega um pacote de comprimento médio de 12000 bits este é segmentado em pacotes menores de 1024, 2048, 3072 e 4096 bits de acordo com a Tabela 2.1. Nessa segmentação o número de *slots* não é inteiro e um dos *slots* será preenchido com zeros, fazendo com que o tamanho do pacote transmitido seja maior do que quando chegou ao sistema.

No caso do sistema IS-856 utilizando FIFO, um pacote é transmitido por inteiro, segmento a segmento, antes de um outro pacote. Isto é, quando um pacote é transmitido, é como se um canal tivesse sido reservado para aquele pacote. Dessa forma, supondo que a chegada de pacotes tenha uma distribuição exponencial negativa com taxa ( $\lambda$ ) e o tempo de atendimento ( $1/\mu$ ) seja também distribuído exponencialmente, o modelo de fila M/M/1 pode ser utilizado para o estudo de desempenho do sistema IS-856. Nesse modelo, o tamanho de pacote não é aquele que chega ao sistema, mas aquele que é transmitido. O cálculo deste tamanho não é fácil e foi determinado empiricamente, através da comparação com os resultados obtidos na simulação.

A Equação 4.1, mostra a expressão encontrada, considerando que o pacote que chega ao sistema tenha a distribuição exponencial de tamanho médio entre  $1000 \leq X \leq 10000$ . Tal equação foi obtida mediante os resultados da simulação na qual foi constatado que o comportamento da curva do tamanho do pacote era similar a uma reta como mostra a Figura 4.18. Desta forma, partindo do princípio que a equação da reta é dada por  $y = ax + b$ . Os coeficientes angular e linear foram aproximados de forma a refletir o comportamento da curva resultando na Equação 4.1. A Figura 4.18 mostra a boa aproximação obtida entre o modelo teórico e a simulação.

$$b = (0,91 + 0,01 \times (X/1000)) \times X + 1478 \times (1 - 0,01 \times (X/1000)) \quad (4.1)$$

Sendo:

X = comprimento médio do pacote poissoniano.

Seja

$$\frac{1}{\mu} = \frac{b}{C} \quad (4.2)$$

Onde:

b = comprimento médio do pacote em bits.

C = capacidade do sistema IS-856 em bits/s.

A capacidade teórica do sistema IS-856 é dada pela soma dos produtos das taxas de dados da

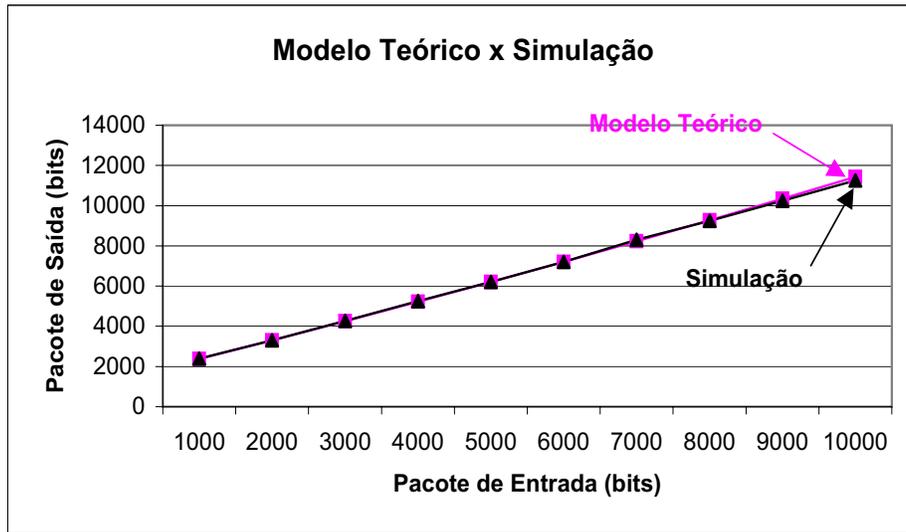


Fig. 4.18: Comparação entre o Modelo Teórico e a Simulação

Tabela 2.1 pela distribuição de probabilidade da Tabela 3.5 tendo como resultado a capacidade de 835000 bits/s. No entanto, devido à segmentação dos pacotes, a taxa efetiva acaba sendo reduzida a um valor menor, e foi encontrada (através da comparação com a simulação) ser em torno de 530000 bits/s. Supondo que tivesse sido adotado o mecanismo de partida lenta do protocolo TCP a vazão do sistema teria inicialmente uma vazão inferior a 530000 bits/s mas conforme a simulação continuasse a vazão tenderia aos 530000 bits/s.

O tempo gasto no sistema é igual a soma do tempo de serviço mais o tempo de espera na fila mostrado pela Equação 4.3.

$$E\{T\} = E\{W\} + E\{X\} \quad (4.3)$$

Onde:

$E\{T\}$  = tempo total gasto no sistema.

$E\{W\}$  = tempo de espera na fila.

$E\{X\}$  = tempo de serviço.

A fim de se comparar o tempo de espera na fila com os resultados obtidos pelas simulações o  $E\{W\}$  foi isolado na Equação 4.4.

$$E\{W\} = E\{T\} - E\{X\} \quad (4.4)$$

Sendo:

$$E\{T\} = \frac{1}{\mu - \lambda} \quad (4.5)$$

$$E\{X\} = \frac{1}{\mu} \quad (4.6)$$

Tem-se:

$$E\{W\} = \frac{1}{\mu - \lambda} - \frac{1}{\mu} \quad (4.7)$$

Onde:

$\lambda$  = taxa de chegada dos pacotes em pacotes/s.

$\mu$  = taxa de atendimento dos pacotes também em pacotes/s.

Por fim, tomando-se como exemplo o escalonamento sem prioridade do tráfego poissoniano SPr Ps do cenário 4 é feita uma comparação entre o tempo de espera na fila da simulação SPr Ps e o tempo de espera na fila do modelo teórico ilustrado na Figura 4.19. Para esta comparação os seguintes parâmetros foram adotados: o tamanho do pacote possui uma distribuição exponencial com média  $X = 6570$  bits porém, o tamanho do pacote segmentado é em torno de  $b = 7990$  bits (Equação 4.1), a capacidade do sistema  $C = 530000$  bits/s e a taxa de chegada dos pacotes  $\lambda$  variando de 20, 40 e 60 pacotes/s para uma ocupação do sistema de 30%, 60% e 90%. Dessa maneira, tem-se que o tempo médio de serviço  $E\{X\} = 15$  ms e o tempo médio de espera na fila  $E\{W\}$  varia de 7 ms a 152 ms muito próximo aos valores obtidos na simulação.

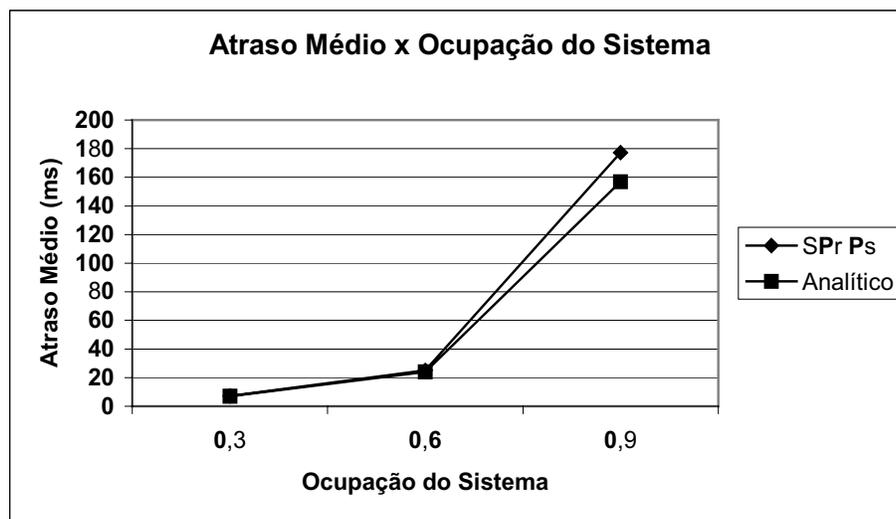


Fig. 4.19: Atraso Médio em função da Ocupação do Sistema com o modelo analítico para o Cenário 4

## 4.8 Conclusão

A maioria dos artigos disponíveis na literatura não aborda a questão relacionada aos atrasos médios dos pacotes, porém aborda outros pontos do sistema IS-856 como a interferência do sinal na interface aérea e a sua vazão. Os resultados obtidos neste estudo mostram que dependendo do escalonador adotado o atraso pode ser um fator que afetará a QoS do sistema. Para garantir uma melhor QoS a priorização de tráfego mostrou-se essencial para os quatro primeiros cenários. As fontes poissonianas apresentam valores superestimados de atrasos quando comparadas com as fontes reais HTTP, FTP e WAP. Entretanto, as fontes poissonianas podem ser utilizadas para estimativas das vazões.

Para os últimos cenários os resultados mostram que dependendo do escalonador adotado o atraso pode ser um fator que afetará a QoS do sistema. Para garantir uma melhor QoS, a reserva de banda dos escalonamentos WFQ e DRR mostra-se essencial em todos os cenários, com o diferencial que além de garantir pequenos atrasos também minimizou os efeitos da perda de pacotes. Já o escalonamento com prioridade mostrou-se adequado apenas para a situação em que há prioridade máxima, porém deixou a desejar os outros tráfegos com prioridade inferiores.

Além disso, estes resultados também mostram que para um tratamento mais justo dos pacotes o escalonamento WFQ revelou-se o mais adequado para tal situação. Por outro lado, o escalonamento DRR permitiu privilegiar tráfegos por meio de uma reserva de banda maior. Desta forma, o escalonamento mais vantajoso dependerá da situação apresentada em que se deve escolher ou não privilegiar determinado tráfego.

# Capítulo 5

## Conclusões

Neste estudo o desempenho do enlace direto do sistema IS-856 foi avaliado através de simulações levando-se em consideração as fontes reais HTTP, WAP e FTP e diversos tipos de escalonamentos FIFO, com prioridade, WFQ e DRR. A vazão, o atraso médio e o percentual de perda dos pacotes foram estudados em função do carregamento do sistema.

Foram feitos estudos do comportamento do enlace direto do sistema IS-856 sob diversos cenários priorizando o tráfego HTTP em relação aos tráfegos FTP e WAP. A priorização de tráfego que não foi objeto de padronização, mostrou-se uma estratégia bastante interessante. A priorização possibilitou um atendimento diferenciado de tráfegos, permitindo, dessa maneira, garantir a qualidade de serviço.

Foi proposto um modelo teórico de desempenho do sistema IS-856 para a estimativa do tempo de espera na fila para escalonamento sem prioridade e com tráfego poissoniano. O modelo proposto quando comparado com a simulação apresentou uma boa aproximação e poderá ser utilizado para a avaliação preliminar do sistema IS-856 quando houver variação de carga.

A avaliação do sistema IS-856 com relação ao atraso médio dos pacotes utilizando as fontes poissonianas resultou sempre em uma estimativa subdimensionada quando comparada com as fontes HTTP e FTP. Entretanto, para avaliação das vazões das fontes HTTP, FTP e WAP, as fontes poissonianas ofereceram boas estimativas.

Através deste estudo, o uso de *buffers* limitados mostrou-se uma boa estratégia para garantir a QoS. Entretanto, para alguns escalonamentos como o sem prioridade a limitação dos *buffers* impôs como consequência altos percentuais de descartes de pacotes o que poderia inviabilizar, por exemplo, a transmissão de fluxos de vídeo.

Por outro lado, os resultados também mostraram que a reserva de banda efetuada pelos escalonamentos WFQ e DRR permitiram assegurar a QoS com pequenos percentuais de perda de pacotes. Além disso, o escalonamento WFQ permitiu um atendimento homogêneo entre os pacotes enquanto o DRR beneficiou os pacotes com maior largura de banda. Portanto, a escolha do melhor escalona-

mento dependerá do quão justo deverá ser o atendimento dos pacotes.

Como futura proposta de trabalho poderia-se implementar o escalonador de equidade proporcional. Este escalonador seria aplicado a cada classe de serviço poderia melhorar ainda mais o desempenho do sistema IS-856.

Outra proposta seria a avaliação de desempenho do enlace reverso do sistema IS-856. Com a implementação deste enlace poderia-se avaliar o desempenho do sistema de uma maneira global por meio dos dois enlaces e desta forma obter medidas de desempenho mais próximas do real.

Além destas propostas, outra interessante seria determinar o dimensionamento de uma rede núcleo que desse suporte ao sistema IS-856 e assim, estimar qual seria o limite de tráfego suportado pelo enlace do sistema IS-856.

# Referências Bibliográficas

- [1] Qualcomm. 1xev: 1x evolution is-856 tia/eia standard airlink overview. White paper, Qualcomm, Novembro 2001. Revision 7.2.
- [2] Andrew S. Tanenbaum. *Computer Networks*, chapter THE PHYSICAL LAYER. PRENTICE HALL, 2002.
- [3] Sandeep Agrawal, Ira Acharya, and Suhel Goel. Inside 3g wireless systems:the 1xev-dv technology. White paper, TATA Consultancy Services, Março 2003. [http://www.tcs.com/0\\_whitepapers/htdocs/EV-DV.pdf/](http://www.tcs.com/0_whitepapers/htdocs/EV-DV.pdf/).
- [4] Hugo Swart. Estudo de desempenho do sistema 1xev-do com prioridades de serviços na interface aérea. Tese de mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Abril 2004.
- [5] Qiang Wu and Eduardo Esteves. *Advances in 3G enhanced technologies for wireless communications*, chapter The cdma2000 High Rate Packet Data System. 2002.
- [6] Hélio Waldman and Michel Daoud Yacoub. *Telecomunicações Princípios e Tendências*. Editora Érica, São Paulo, 1 edition, 1997.
- [7] Eduardo Tude. Gprs. Tutorial, Novembro 2003. <http://www.teleco.com.br/tutoriais/tutorialgprs/>.
- [8] Michel Daoud Yacoub. *Wireless Technology Protocols, Standards, and Techniques*. CRC, EUA, 2001.
- [9] Marilson Duarte Soares. Edge. Tutorial, Novembro 2003. <http://www.teleco.com.br/tutoriais/tutorialedge/>.
- [10] 3GPP2. cdma2000, high rate packet data air interface specification. Technical Report C.S0024, Third Generation Partnership Project 2, Outubro 2002. <http://www.3gpp2.org/>.

- [11] Qi Bi. A forward link performance study of the 1xev-do system through simulations and field measurements. Março 2004.
- [12] Hugo Swart and Shusaburo Motoyama. Estudo de desempenho do sistema 1xev-do com prioridades de serviços na interface aérea. *Simpósio Brasileiro de Telecomunicações*, Outubro 2003. Rio de Janeiro.
- [13] Qi Bi and Stan Vitebsky. Performance analysis of 3g 1xev-do high data rate system. *IEEE Wireless Communications and Networking Conference*, 1:389–395, Março 2002. EUA.
- [14] Kuenyoung Kim, Hoon Kim, and Youngnam Han. A proportionally fair scheduling algorithm with qos and priority in 1xev-do. *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 5:2239–2243, Setembro 2002. Portugal.
- [15] Qi Bi, Ronald R. Brown, Dongzhe Cui, Asif D. Gandhi, Ching-Yao Huang, and Stan Vitebsky. Performance of 1xev-do third-generation wireless high-speed data systems. *Bell Labs Technical Journal*, 2003.
- [16] Young-June Choi and Saewoong Bahk. Qos sheduling for multimedia traffic in packet data cellular networks. *IEEE International Conference on Communications*, 1:358–362, Maio 2003. EUA.
- [17] Mingxi Fan and Eduardo Esteves. The impact of antenna-array receivers on the reverse link performance of cdma 2000 1xev high rate packet data systems. *IEEE Vehicular Technology Conference*, 2:841–846, Outubro 2003. EUA.
- [18] ChingYao Huang, Qi Bi, Asif Gandhi, Ron Brown, and Dongzhe Cui. Forward and reverse link capacity for 1xev-do: Third generation wireless high-speed data systems. *IEEE Global Telecommunications Conference*, 1:871–875, Novembro 2002. Taiwan.
- [19] Eduardo Esteves, Mehmet I. Gurelli, and Mingxi Fan. Performance of fixed wireless access with cdma2000 1xev-do. *IEEE Vehicular Technology Conference*, 2:836–840, Outubro 2003. EUA.
- [20] 3GPP2 WG5 Evaluation Ad Hoc. 1xev-dv evaluation methodology - addendum (v6). Technical report, Third Generation Partnership Project 2, Julho 2001.
- [21] Paul Bender, Peter Black, Matthew Grob, Roberto Padovani, Nagabhushana Sindhushayana, and Andrew Viterbi. Cdma/hdr: A bandwidth efficient high speed wireless data service for nomadic users. *IEEE Communications Magazine*, 38:70–77, Julho 2000.

- [22] Qualcomm. 1xev-do system architecture. White paper, Qualcomm, 2003.
- [23] Qualcomm. 1xev-do session protocol stacks. White paper, Qualcomm, 2003.
- [24] Nagabhushana T. Sindhushayana and Peter J. Black. Forward link coding and modulation for cdma2000 1xev-do (is-856). *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 4:1839 – 1846, Setembro 2002.
- [25] Eduardo Esteves. On the reverse link capacity of cdma2000 high rate packet data systems. *IEEE International Conference on Communications*, 3:1823 – 1828, Abril 2002. EUA.
- [26] Hyoung-Kee Choi and John O. Limb. A behavioral model of web traffic. *Seventh International Conference on Network Protocols*, pages 327–334, Novembro 1999. Canadá.
- [27] Merran Evans, Nicholas Hastings, and Brian Peacock. *Statistical Distributions*. John Wiley and Sons, EUA, 2 edition, 1993.
- [28] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 2:137 – 150, Abril 1993.
- [29] M. Shreedhar and George Varghese. Efficient fair queuing using deficit round-robin. *IEEE/ACM Transactions on Networking*, 4:375 – 385, Junho 1996.
- [30] Shusaburo Motoyama. Fila de multi-usuários sem e com prioridade. Apostila do curso ie670 aula 11, DT - Unicamp, 2005. <http://www.dt.fee.unicamp.br/motoyama/ie670/ie670.html>.

# Apêndice A

Neste apêndice são apresentadas comparações do modelo simulado do sistema IS-856 proposto neste trabalho com o modelo teórico proposto em [4]. O objetivo dessas comparações é verificar a consistência do sistema IS-856 desenvolvido em Matlab. São apresentados, também, histogramas das fontes poissonianas, desenvolvidas em Matlab.

## A.1 Consistência do Modelo IS-856 Desenvolvido

Nesta seção os resultados das simulações do modelo IS-856 são comparados com os resultados teóricos do cenário 1 proposto em [4]. Neste cenário o tráfego de pacotes poissoniano é dividido em prioridades conforme mostra a Tabela A.1. Cada prioridade desta tabela é distribuída conforme a coluna Probabilidade (Pr).

Tab. A.1: Distribuição *Default* das Prioridades (Pr)

Prioridade	Probabilidade (Pr)
1	10%
2	25%
3	30%
4	35%

O número de usuários na célula, denominado N, varia de 5 usuários até 20 usuários. As classes de cobertura estão divididas em proporções (Probabilidade) conforme mostra a Tabela 3.5.

A vazão média por usuário é dada por  $R_u$  e o valor padrão considerado por usuário é  $R_u = 17$  kbps. Para se calcular a taxa média de chegada de pacotes ( $\lambda$ ) de cada classe de cada prioridade é utilizada a expressão A.1.

$$\lambda_x = \frac{N * Pr * Probabilidade * R_u}{T_c} \quad (\text{A.1})$$

Onde  $N$ ,  $Pr$ ,  $Probabilidade$ ,  $R_u$  foram parâmetros definidos anteriormente e  $T_c$  representa o comprimento de pacote em bits de cada classe de cobertura.

O modelo teórico aplicado a este sistema é o modelo de fila  $M/G/1$  de multi-usuários com prioridade. Este modelo permite o cálculo do tempo médio de espera na fila de cada prioridade por meio da expressão A.2.

$$E\{W_p\} = \frac{E\{T_0\}}{(1 - \sigma_{p-1})(1 - \sigma_p)} \quad p = 2, \dots, r \quad (\text{A.2})$$

Onde  $\sigma_p$  é definido por:

$$\sigma_0 = 0, \sigma_1 = \rho_1, \sigma_2 = \rho_1 + \rho_2, \dots, \sigma_r = \rho \quad (\text{A.3})$$

$\rho_k$  representa a carga de cada usuário ou classe de transmissão e dado por:

$$\rho_k = \frac{\lambda_k}{\mu_k} \quad (\text{A.4})$$

Onde  $\lambda_k$  é a taxa de chegada de cada classe e  $\mu_k$  é a taxa de atendimento de cada classe.

$\rho$  é a ocupação total do sistema, soma dos recursos que cada usuário consome do sistema.

$$\rho = \sum_{k=1}^r \rho_k \quad (\text{A.5})$$

$E\{T_0\}$  é o tempo residual de atendimento quando da chegada de um novo pacote e sua esperança é dada pela equação A.6.

$$E\{T_0\} = \frac{1}{2} \sum_{k=1}^r \lambda_k E\{R_k^2\} \quad (\text{A.6})$$

$E\{R_k^2\}$  é o valor quadrático médio do tempo de serviço.

Para a prioridade mais alta a equação A.7 é dada por:

$$E\{W_1\} = \frac{E\{T_0\}}{1 - \rho_1} \quad (\text{A.7})$$

Para a menor prioridade a expressão A.8 se resume a:

$$E\{W_p\} = \frac{E\{T_0\}}{(1 - \sigma_{p-1})(1 - \rho)} \quad (\text{A.8})$$

Para o cálculo do atraso médio dos pacotes na fila sem prioridade é utilizada a equação A.9.

$$E\{W\} = \frac{E\{T_0\}}{(1 - \rho)} \quad (\text{A.9})$$

Maiores detalhes sobre a fila M/G/1 podem ser obtidos em [30].

Por fim, para o cálculo da vazão do sistema com prioridade é usada a seguinte expressão A.10.

$$R_p = N * R_u * Probabilidade \quad (\text{A.10})$$

Para o cálculo da vazão sem prioridade é usada a expressão A.11.

$$R = N * R_u \quad (\text{A.11})$$

Desta forma, fazendo uso deste modelo teórico são feitas comparações entre os resultados simulados e resultados teóricos nas Figuras A.1, A.2, A.3, A.4, A.5 e A.6. As comparações mostram que os resultados das simulações estão coerentes com os resultados teóricos.

Nas Figuras A.1, A.2 e A.3 são feitas comparações entre os modelos teóricos e simulados levando-se em consideração um sistema sem prioridade.

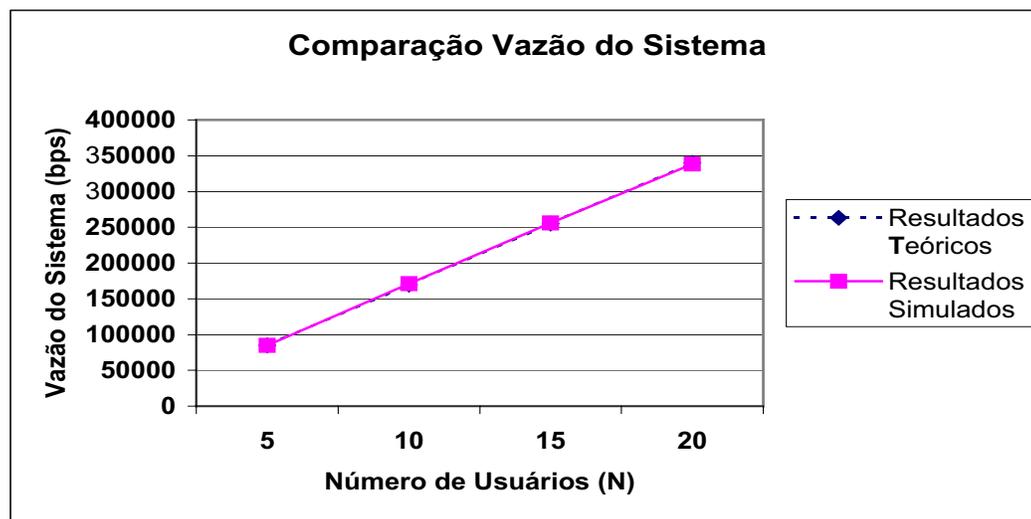


Fig. A.1: Vazão em Função do Número de Usuários

Já nas Figuras A.4, A.5 e A.6 são feitas comparações entre os modelos teóricos e simulados levando-se em consideração um sistema com prioridade.

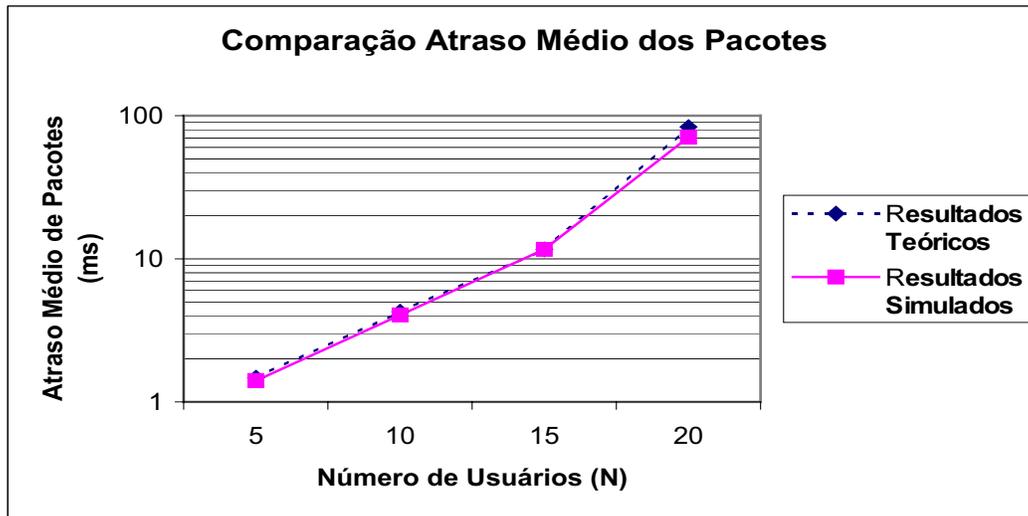


Fig. A.2: Atraso Médio dos Pacotes em Função do Número de Usuários

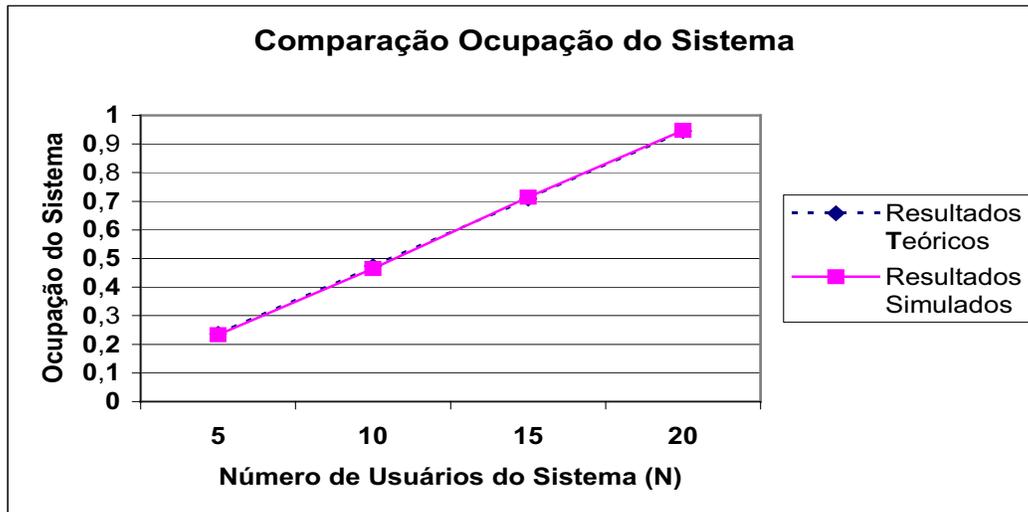


Fig. A.3: Fator de Utilização do Sistema Versus o Número de Usuários

## A.2 Histogramas das Fontes Poissonianas

A Figura A.7 mostra o histograma do comprimento de pacotes obtido na simulação. A média dos pacotes é de 1500 bytes, e esta média é identificada pelos asteriscos presentes no gráfico. Pode-se observar que o gráfico obtido é aproximadamente uma exponencial negativa. Nesta figura também pode ser observado um salto devido a um limite do intervalo de pacotes que podem variar de 0 a 6000 pacotes. Se não houvesse este limite de intervalo a curva do gráfico continuaria a sua tendência de

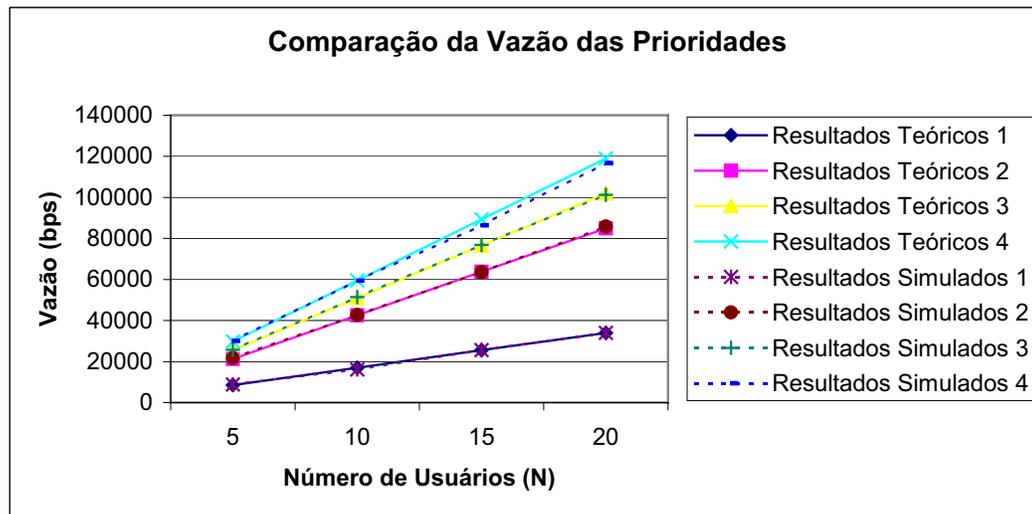


Fig. A.4: Vazão em Função do Número de Usuários

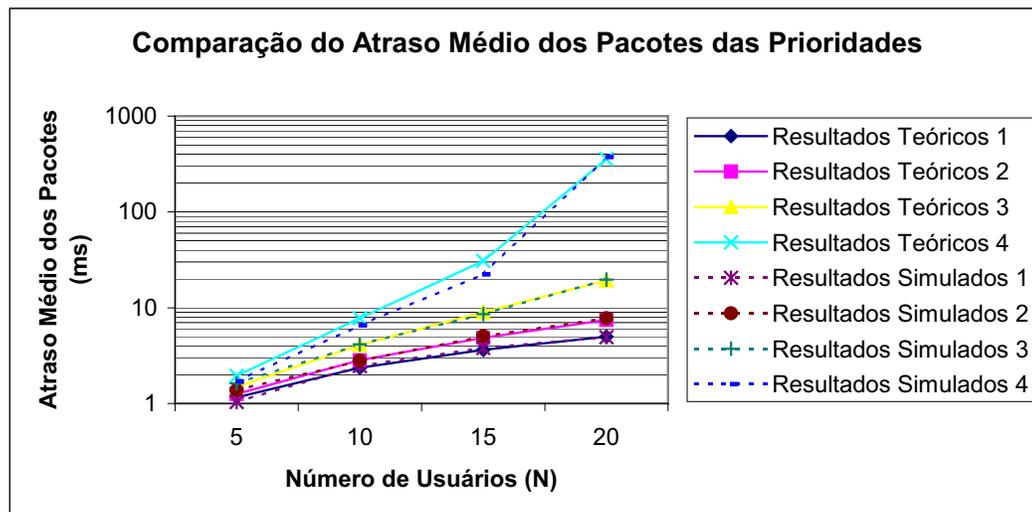


Fig. A.5: Atraso Médio dos Pacotes em Função do Número de Usuários

chegar a zero.

Na Figura A.8 é exibido o processo de chegada do tráfego poissoniano em função da ordem de chegada dos pacotes no tempo. Por exemplo, o pacote de ordem 2000 chegou no instante 120 s, o pacote de ordem 4000 chegou no instante 200 s, ou seja, um único pacote chega por instante de tempo e não 4000 pacotes chegam no instante 200 s.

A Figura A.9 mostra um histograma com distribuição exponencial com média de 50 ms do intervalo (em tempo) entre chegadas de pacotes do tráfego poissoniano.

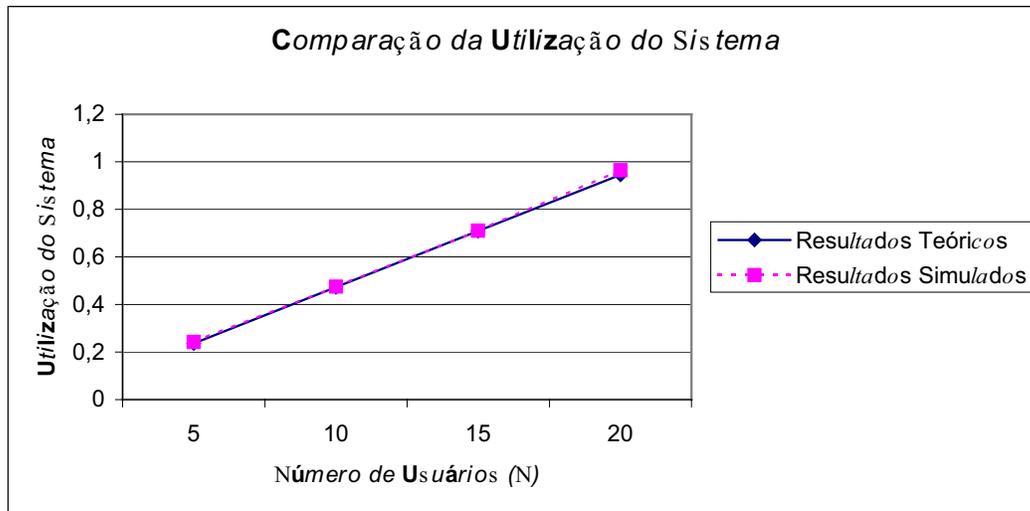


Fig. A.6: Fator de Utilização do Sistema Versus o Número de Usuários

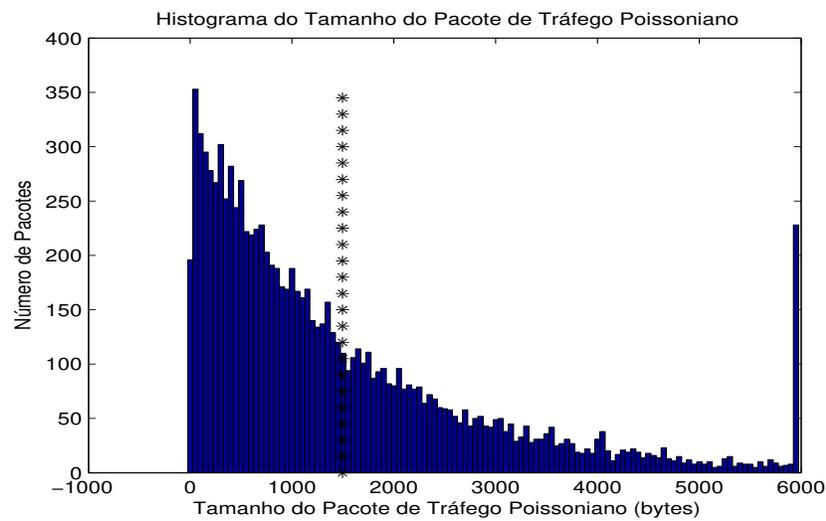


Fig. A.7: Histograma do Comprimento do Pacote do Tráfego Poissoniano

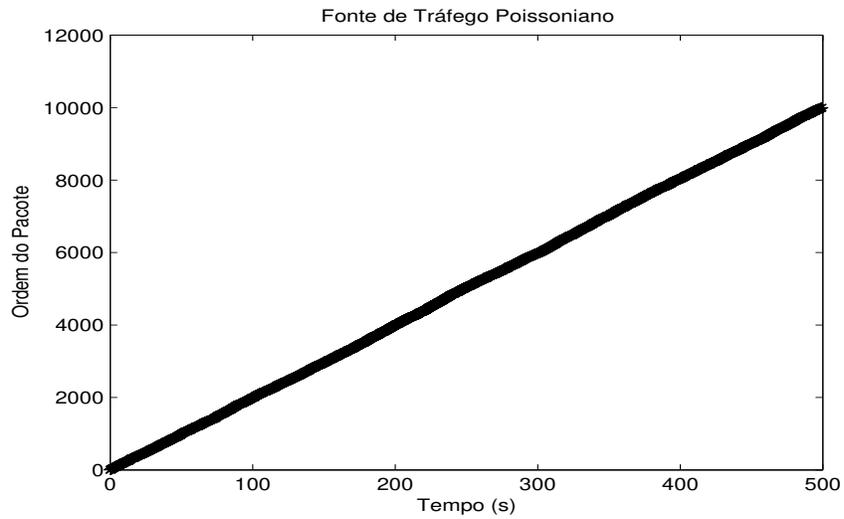


Fig. A.8: Ordem de Chegada dos Pacotes do Tráfego Poissoniano

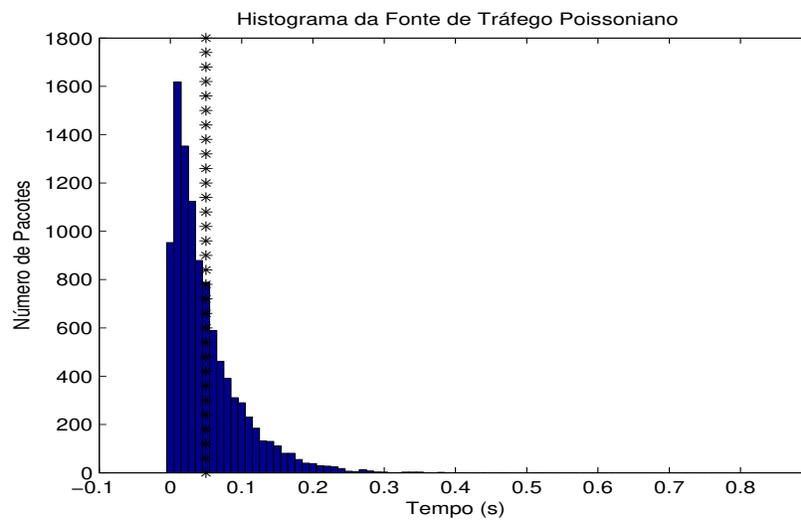


Fig. A.9: Histograma dos Intervalos entre Chegadas do Tráfego Poissoniano

# Apêndice B

Neste apêndice são apresentados os códigos fontes em Matlab das fontes HTTP, FTP e WAP avaliadas além do fluxograma do simulador utilizado neste trabalho.

## B.1 Programa Principal

A Figura B.1 mostra o fluxograma do programa principal do simulador adotado neste trabalho.

No início do programa ocorre a inicialização das variáveis. A variável relógio representa o tempo simulado no programa sendo inicializada com zero. O número de pacotes na fila é representado pela variável fila e inicialmente é zerada. Por fim, a variável livre indica o estado do servidor em que no início do programa é setada em um mostrando que o servidor está pronto para atender o pacote.

O simulador desenvolvido neste trabalho caracteriza-se por ser um simulador de eventos discretos em que podem ocorrer dois tipos de eventos: chegada de pacote ou término de serviço do pacote. Este dois eventos estão representados no fluxograma pelas variáveis chegada\_pacote e término\_serviço. No início do programa o único evento possível de ocorrer é chegada do primeiro pacote ao sistema. Desta forma, a variável chegada\_pacote recebe o tempo de chegada do primeiro pacote e como este é único evento possível término\_serviço e chegada\_pacote são igualados.

A cada iteração do programa verifica-se qual é o evento iminente e atualiza-se o relógio. Se o evento for de chegada de pacote é verificado se o servidor está livre. Estando livre, o servidor muda de estado e passa a estar ocupado e em seguida o pacote é atendido. Além disso, a variável chegada\_pacote recebe o tempo de chegada do próximo pacote a chegar no sistema. Senão se o servidor estiver ocupado o pacote é armazenado na fila para atendimento posterior.

Caso o evento seja de término de serviço verifica-se a fila. Se a fila estiver vazia o servidor muda de estado setando a variável livre em um e aguarda o próximo evento a ocorrer no sistema, como não há nenhum pacote sendo atendido no sistema o próximo evento será obrigatoriamente a chegada de um novo pacote. Senão caso exista algum pacote na fila decrementa-se o número de pacotes na fila e o atende-se este pacote.

A condição de saída do *loop* do programa principal se dá no momento em que o último pacote de alguma das fontes citadas nos cenários descritos no Capítulo 3 for processado.

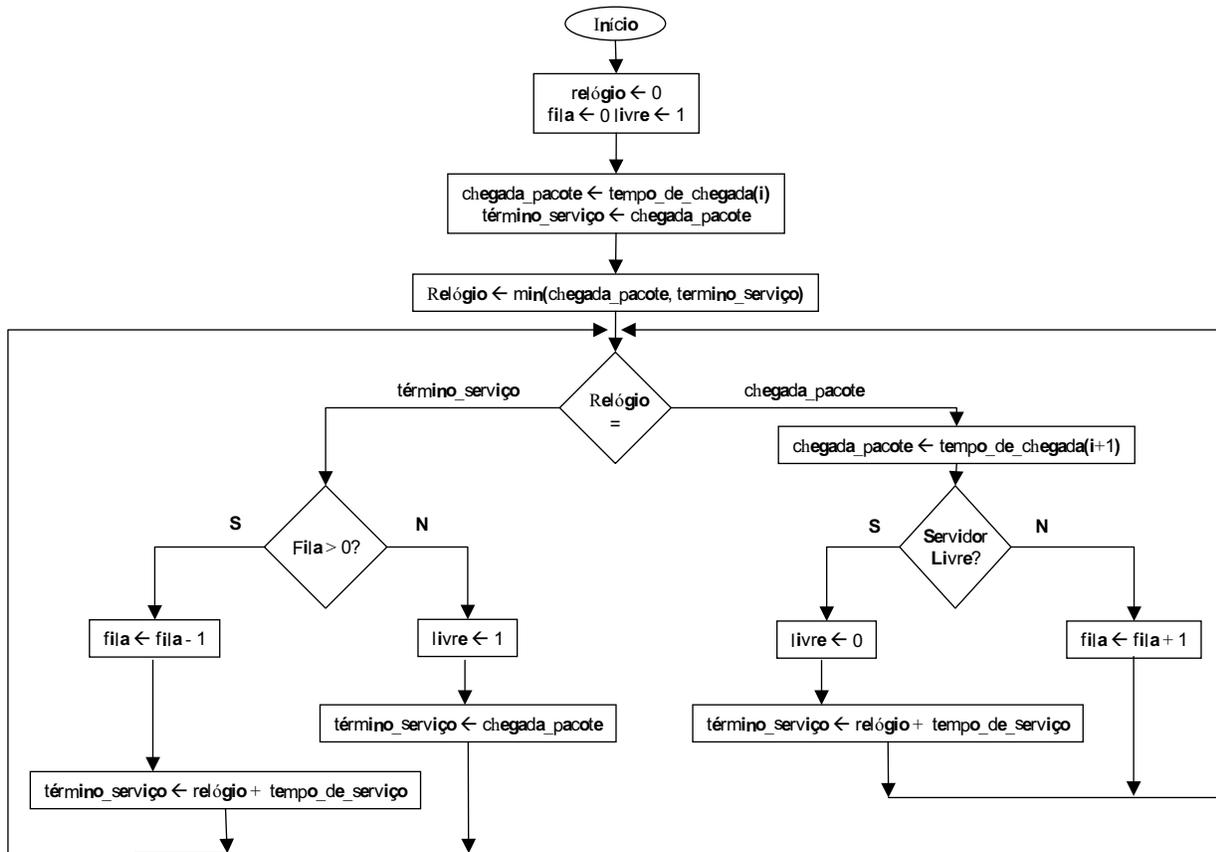


Fig. B.1: Fluxograma do Programa Principal do Simulador

## B.2 Fonte HTTP

```

% Fonte_HTTP.m
% Data: 03/08/04

clc clear format short g

Max = 1000;

tempo_de_chegada = zeros(1,Max);
tamanho = zeros(1,Max); % Tamanho do pacote

On = 1; % estado ativo

lambda1 = 0.033; % Taxa do tempo de leitura
  
```

```

MTU = 0; % 76% das vezes o tamanho maximo do pacote sera 1500 bytes e em 24% das vezes 576 bytes

i = 1;

Max_Local = 0;

b1 = waitbar(0,'Simulando fonte HTTP'); while i < Max
    if On == 1

        sorteio = rand;
        if sorteio <= 0.76
            MTU = 1500;
            MTU_40 = 1460;
        else
            MTU = 576;
            MTU_40 = 536;
        end

        sorteio = rand;
        if sorteio <= 0.5
            Max_Local = Max_Local + 1000;
            HTTP_Burst;
        else
            Max_Local = Max_Local + 1000;
            HTTP_Persistent;
        end

        On = 0;
    else
        tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda1;
        On = 1;
        i = i + 1;
    end
    waitbar(i/Max,b1)
end

close(b1)

```

### B.2.1 HTTP Burst 1.0

```

% Simula uma fonte HTTP 1.0 Burst Mode
% Data: 03/08/04

On = 1; % estado ativo

lambda2 = 0.033;% Tempo de Leitura
lambda3 = 7.69; % Taxa parsing Time
lambda4 = 20; % Tempo de propagacao

% Parametros da distribuicao de Pareto
k = 1.75; alfa = 1.1;

```

```

while i < Max_Local
    if On == 1

        % Tamanho do objeto principal segundo a distribuicao lognormal com media 10710 bytes
        tamanho_objeto_principal = round(lognrnd(8.35,1.37));
        % Enquanto o tamanho do arquivo for maior que 2 MB e menor que 100 bytes gere outro tamanho aleatorio
        while (tamanho_objeto_principal > 2097152) | (tamanho_objeto_principal < 100)
            tamanho_objeto_principal = round(lognrnd(8.35,1.37));
        end

        % Envio de segmentos SYN+ACK e segmentos de controle
        tamanho(i) = 80;

        % Tempo de propagacao
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
        i = i + 1;

        % Calcula-se o numero de pacotes necessarios para transmitir o arquivo
        n_pacotes = tamanho_objeto_principal/MTU_40;

        P = fix(n_pacotes);

        for j = 1:(P + 1)
            % Tempo de propagacao
            tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda4;
            tamanho(i) = MTU;
            i = i + 1;
        end

        % Envio do segmento FIN e do segmento ACK, fecha a conexao
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda1;
        tamanho(i) = 80;
        i = i + 1;

        % Parsing Time
        tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda3;
        i = i + 1;

        % Numero de objetos embutidos segundo a distribuicao de pareto com media = 5.64
        n_objetos = round(k*(1 - rand)^(-1/alfa)); % Numero de objetos embutidos
        % Enquanto o numero de objetos embutidos for maior do que 53 gera-se outro numero aleatorio
        while n_objetos > 53
            n_objetos = round(k*(1 - rand)^(-1/alfa));
        end

        resto_composto = 0;

        if (n_objetos/4) <= 1
            b = n_objetos;
            n_compostos = 1;
        else
            b = 4;
            n_compostos = fix(n_objetos/4);
            if ((n_objetos/4) - fix(n_objetos/4))*4 > 0

```

```
        resto_composto = ((n_objetos/4) - fix(n_objetos/4))*4;
    end
end

for il = 1:n_compostos
    objeto_composto = 0;
    for w = 1:b
        % Tamanho do objeto embutido segundo a distribuicao lognormal com media 7758 bytes
        tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
        % Enquanto o tamanho do objeto embutido for maior que 2 MB e menor que 50 bytes gere outro tamanho
        while (tamanho_objeto_embutido > 2097152) | (tamanho_objeto_embutido < 50)
            tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
        end

        objeto_composto = objeto_composto + tamanho_objeto_embutido;
    end

    % Envio de segmentos SYN+ACK e segmentos de controle
    tamanho(i) = 80*b;
    % Tempo de propagacao
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
    i = i + 1;

    % Calcula-se o numero de pacotes necessarios para transmitir o arquivo
    n_pacotes = objeto_composto/MTU_40;

    P = fix(n_pacotes);

    for j = 1:(P + 1)
        tamanho(i) = MTU;
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
        i = i + 1;
    end

    tamanho(i) = 80*b;
    % Tempo de propagacao
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
    i = i + 1;
end % fim for

if resto_composto > 0
    for w = 1:resto_composto
        % Tamanho do objeto embutido segundo a distribuicao lognormal com media 7758 bytes
        tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
        % Enquanto o tamanho do objeto embutido for maior que 2 MB e menor que 50 bytes gere outro tamanho
        while (tamanho_objeto_embutido > 2097152) | (tamanho_objeto_embutido < 50)
            tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
        end

        objeto_composto = objeto_composto + tamanho_objeto_embutido;
    end

    % Envio de segmentos SYN+ACK e segmentos de controle
    tamanho(i) = 80*resto_composto;
```

```

% Tempo de propagacao
tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
i = i + 1;

% Calcula-se o numero de pacotes necessarios para transmitir o arquivo
n_pacotes = objeto_composto/MTU_40;

P = fix(n_pacotes);

for j = 1:(P + 1)
    tamanho(i) = MTU;
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
    i = i + 1;
end

tamanho(i) = 80 * resto_composto;
% Tempo de propagacao
tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
i = i + 1;

end % fim if resto_composto
On = 0;
else % fim if On == 1
    break;
end
end % fim while

```

## B.2.2 HTTP Persistente 1.1

```

% Simula uma fonte HTTP 1.1 Persistent Mode
% Data: 03/08/04

On = 1; % estado ativo

lambda2 = 0.033;% Tempo de leitura
lambda3 = 7.69; % Taxa parsing Time
lambda4 = 20; % Tempo de propagacao

% Parametros da distribuicao de Pareto
k = 1.75; alfa = 1.1;

while i < Max_Local
    if On == 1
        % Envio de segmentos SYN+ACK e segmentos de controle
        tamanho(i) = 80;

        % Tempo de propagacao
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
        i = i + 1;

        % Tamanho do objeto principal segundo a distribuicao lognormal com media 10710 bytes
        tamanho_objeto_principal = round(lognrnd(8.35,1.37));
        % Enquanto o tamanho do arquivo for maior que 2 MB e menor que 100 bytes gere outro tamanho aleatorio
    end
end

```

```
while (tamanho_objeto_principal > 2097152) | (tamanho_objeto_principal < 100)
    tamanho_objeto_principal = round(lognrnd(8.35,1.37));
end

% Calcula-se o numero de pacotes necessarios para transmitir o arquivo
n_pacotes = tamanho_objeto_principal/MTU_40;

P = fix(n_pacotes);

% Transferencia do objeto
% Se o numero de pacotes for maior do que 1
for j = 1:(P + 1)
    % Tempo de propagacao
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
    tamanho(i) = MTU;
    i = i + 1;
end

% Parsing Time
tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda3;
i = i + 1;

% Numero de objetos embutidos segundo a distribuicao de pareto com media = 5.64
n_objetos = round(k*(1 - rand)^(-1/alfa)); % Numero de objetos embutidos
% Enquanto o numero de objetos embutidos for maior do que 53 gera-se outro numero aleatorio
while n_objetos > 53
    n_objetos = round(k*(1 - rand)^(-1/alfa));
end

% Transmite 4 pacotes de 40 bytes
tamanho(i) = 160;

Nd = 1;

while Nd <= n_objetos
    % Tempo de propagacao
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
    i = i + 1;

    % Tamanho do objeto embutido segundo a distribuicao lognormal com media 7758 bytes
    tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
    % Enquanto o tamanho do objeto embutido for maior que 2 MB e menor que 50 bytes gere outro tamanho al
    while (tamanho_objeto_embutido > 2097152) | (tamanho_objeto_embutido < 50)
        tamanho_objeto_embutido = round(lognrnd(6.17,2.36));
    end

    % Calcula-se o numero de pacotes necessarios para transmitir o arquivo
    n_pacotes = tamanho_objeto_embutido/MTU_40;

    P = fix(n_pacotes);

    for j = 1:(P + 1)
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
```

```

        tamanho(i) = MTU;
    end

    Nd = Nd + 1;
end

tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda4;
% Envio do segmento FIN e do segmento ACK, fecha a conexao
tamanho(i) = 80;
i = i + 1;

On = 0;
else % fim if On == 1
    break;
end
end % fim while

```

## B.3 Fonte FTP

```

% Simula uma fonte FTP
% Data: 03/08/04
Max = 5000;

tempo_de_chegada = zeros(1,Max);
tamanho = zeros(1,Max); % Tamanho do pacote

On = 1; % estado ativo

lambda1 = 0.006; % Tempo de leitura
lambda2 = 20; % Tempo de propagacao

MTU1 = 1500; % 76% Tamanho maximo do pacote em bytes
MTU1_40 = 1460;
MTU2 = 576; % 24 % Tamanho maximo do pacote em bytes
MTU2_40 = 536;

i = 1;

b1 = waitbar(0,'Simulando fonte FTP'); while i < Max
    if On == 1
        % Tamanho do arquivo segundo a distribuicao lognormal com media 2 MB = 2097152 bytes
        tamanho_do_arquivo = round(lognrnd(14.45,0.35));
        % Enquanto o tamanho do arquivo for maior que 5 MB gere outro tamanho aleatorio
        while tamanho_do_arquivo > 5242880 % == ((1024)^2)*5 == 5242880 bytes
            tamanho_do_arquivo = round(lognrnd(14.45,0.35));
        end

        % 76% das vezes o MTU sera 12000 bits ou 1500 bytes
        sorteio = rand;
        if sorteio <= 0.76

```

```

% Calcula o numero de pacotes necessarios para transmitir o arquivo
n_pacotes = tamanho_do_arquivo/MTU1_40;

P = fix(n_pacotes);

% Se o numero de pacotes for maior do que 1
for j = 1:(P + 1)
    tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda2;
    tamanho(i) = MTU1;
    i = i + 1;
end
else % Em 24% das vezes o MTU sera 4608 bits ou 576 bytes

    % Calcula o numero de pacotes necessarios para transmitir o arquivo
    n_pacotes = tamanho_do_arquivo/MTU2_40;

    P = fix(n_pacotes);

    for j = 1:(P + 1)
        % Tempo de propagacao
        tempo_de_chegada(i + 1) = tempo_de_chegada(i) + -log(rand)/lambda2;
        tamanho(i) = MTU2;
        i = i + 1;
    end
end
On = 0;
else % Senao estado Off
    % Tempo de leitura entre os arquivos com distribuicao poissoniana com media 180 s
    tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda1;
    On = 1;
    i = i + 1;
end
waitbar(i/Max)
end
close(b1);

```

## B.4 Fonte WAP

```

% Simula uma fonte WAP
% Data: 10/08/04

Max = 1000;

tempo_de_chegada = zeros(1,Max);

tamanho_pacote_wap = zeros(1,Max);

lambda1 = .625; % Tempo inter-chegada entre objetos
lambda2 = .4; % Tempo de resposta do gateway WAP
lambda3 = .182; % Tempo de leitura

```

```
% Parametros da distribuicao de Pareto
k = 93.21; alfa = 1.1;

i = 1;

On = 1; % estado ativo

b1 = waitbar(0,'Simulando fonte WAP'); while i < Max
    if On == 1
        % Tempo de resposta do gateway WAP
        tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda2;
        i = i + 1;

        % Numero de objetos por resposta distribuido geometricamente com media = 2
        n_objetos = round(geornd(1/3));

        % Enquanto numero de objetos for igual a 0 gera-se outro numero aleatorio
        while n_objetos < 1
            n_objetos = round(geornd(1/3));
        end

        for j = 1:n_objetos
            % Geracao do tamanho_pacote_wap do objeto segundo a distribuicao de pareto com media 256 bytes
            n_aleatorio = round(k*(1 - rand)^(-1/alfa));

            % Enquanto o tamanho_pacote_wap do objeto for maior do que 1400 bytes gera-se outro numero aleatorio
            while n_aleatorio > 1400
                n_aleatorio = round(k*(1 - rand)^(-1/alfa));
            end

            tamanho_pacote_wap(i) = n_aleatorio;

            if j < n_objetos
                % Tempo inter-chegadas entre os objetos segundo a distribuicao poissoniana com media 1.6 s
                tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda1;
                i = i + 1;
            end
        end

        end
        On = 0;
    else % Senao estado Off
        % Tempo de leitura
        tempo_de_chegada(i+1) = tempo_de_chegada(i) + -log(rand)/lambda3;
        On = 1;
        i = i + 1;
    end

    waitbar(i/Max,b1)
end
close(b1);
```